



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Kerja praktek magang dilakukan pada *Department Research and Development Center* yang dikepalai oleh Bapak Bambang Leksono. Pada Awalnya melaksanakan kerja magang sebagai *System Analyst* yang berada dibawah pengawasan Bapak Petrus Andhy Prabowo untuk mengerjakan proyek *chatbot* berbasis *python*, namun dipindahtugaskan sebagai *Front-End Developer* untuk mengerjakan proyek Aplikasi Pocketbank Agent yang berada dibawah pengawasan Ibu Rahajeng Iga Titisari selaku *project manager*. Selama pengembangan aplikasi dibantu oleh Bapak Ahmad Fahryan dan Ibu Runy Eka selaku *team lead*, bersama dengan empat peserta magang dari universitas lain.

3.2. Tugas yang Dilakukan

Tugas yang dilakukan adalah membangun *web service* pada *chatbot* sebagai *system analyst*. Karena terdapat masalah internal pada perusahaan, maka terjadi pemindahan tugas yang dilakukan, yaitu sebagai *front-end developer* untuk membuat Aplikasi Pocketbank Agent dengan menggunakan React Js yang merupakan *library* dari JavaScript khususnya pada fitur *Change Language*, *Change Password*, *Change PIN*, *Transaction* dan *List Customer Registration*. Penjabaran tugas yang dilakukan setiap minggu dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Deskripsi Tugas yang Dilakukan per Minggu

Minggu ke-	Jenis Pekerjaan yang Dilakukan
1	Mempelajari tentang <i>web service</i>
	Mempelajari tentang RESTful API pada <i>web service</i>
2	Mempelajari cara membuat <i>simple web services description language</i>
3	Membuat <i>database</i> untuk <i>login service</i>
4	Membuat <i>login service administrator</i>
5	Membuat <i>login service administrator</i>
6	Mempelajari React Native
	Mempelajari <i>base project</i> Pocketbank Agent berbasis web dengan menggunakan React Js
7	Membuat fitur <i>change language</i> pada Aplikasi Pocketbank Agent
8	Membuat fitur <i>change password</i> pada Aplikasi Pocketbank Agent
9	Membuat fitur <i>postpaid electricity form</i> dan fitur <i>customer view detail</i>
10	Membuat fitur <i>transaction, cash in form, cash in modal confirm, cash in modal receipt, dan cash in function print</i>
11	Membuat fitur <i>cash out form, cash out modal confirm, cash out modal PIN, cash out modal receipt, dan cash out function print</i>
12	Membuat fitur <i>forgot PIN, change PIN, dan activation account page</i>
	Memperbaiki <i>bug</i> dan <i>validation</i>

3.3. Uraian Kerja Magang

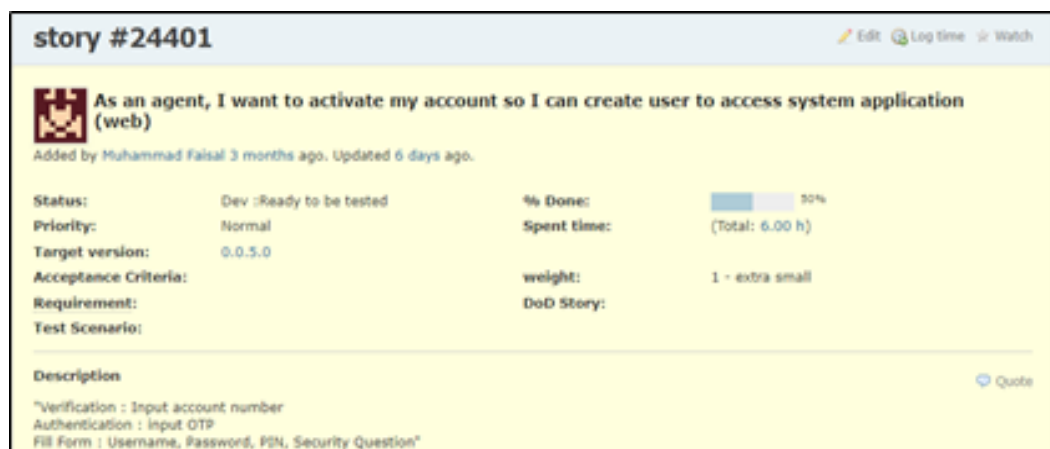
Proses pelaksanaan kerja magang dimulai dengan pelatihan mengenai bahasa pemrograman yang pada awalnya menggunakan *python* dan *tool* yang digunakan untuk mengembangkan modul *web service* dan hal itu berlaku juga saat

pemindahan tugas pelaksanaan kerja magang yang selalu dimulai dengan pelatihan mengenai bahasa pemrograman dan semua *framework* dan *tools* yang digunakan. Selama pelatihan, setiap peserta magang diminta untuk membuat sebuah proyek sederhana yang ditentukan dan mencakup semua materi yang telah dipelajari dan mempresentasikan proyek yang telah dibuat pada akhir pelatihan (biasanya pada minggu berikutnya). Pelatihan yang dilakukan secara individual dan otodidak sesuai dengan materi pembelajaran yang telah diberikan oleh pembimbing lapangan. Tabel 3.2 menunjukkan materi yang dipelajari selama pelatihan kerja magang.

Tabel 3.2 Materi Pembelajaran Terkait Magang

Topik Pembelajaran	Keterangan
Instalasi	<i>Web Service</i> <ul style="list-style-type: none"> • Dapat menginstal Anaconda Navigator • Dapat menginstal Python • Dapat menginstal PostgreSQL • Dapat menginstal Linux • Dapat menjalankan <i>syntax</i> DDL sederhana
	Pocketbank Agent <ul style="list-style-type: none"> • Dapat menginstal npm atau yarn • Dapat menginstal SDK untuk React Native • Dapat menginstal React Native • Memahami struktur react-redux • Memahami penggunaan Gitlab • Dapat menginstal Visual Studio Code
<i>Hardware</i> yang digunakan	<ul style="list-style-type: none"> • <i>Processor</i> Intel Core i5-6200U • <i>Memory</i> 12 GB DDR 4 • Sistem Operasi Microsoft Windows 10 Pro (64-bit) • Hard Disk 1 TB
React Component	<ul style="list-style-type: none"> • Memahami tentang <i>base project</i>, struktur <i>page</i>, <i>action</i>, <i>reducer</i>, dan <i>component</i>
Redmine	Memahami penggunaan aplikasi manajemen proyek

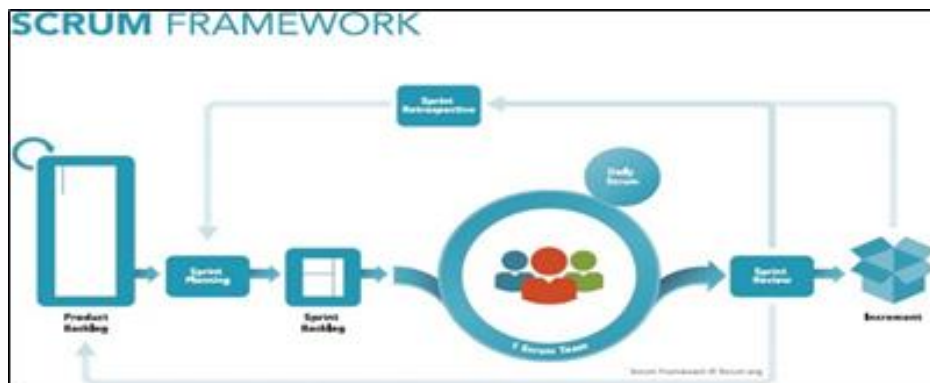
Selama pelaksanaan kerja magang, setiap proyek yang dikembangkan dengan menggunakan model pengembangan *software scrum*. *Scrum* merupakan sebuah metode rekayasa perangkat lunak dengan menggunakan prinsip *agile* yang bertumpu pada kekuatan kolaborasi dengan *team*, *incremental product*, dan proses iterasi untuk mewujudkan hasil akhir dimana setiap orang dapat mengatasi masalah yang kompleks sambil menghasilkan produk dengan nilai tertinggi secara produktif (Schwaber dan Sutherland, 2017). Teknik pengembangan *software scrum* biasanya dimulai dari pembahasan *project* antara *product owner* dengan *stakeholder* yang terkait dan dibentuk *scrum master* (*project manager*) dan *team development*. Pada awalnya *scrum master* membuat semua *list* pekerjaan yang disebut *user story*. Gambar 3.1 menunjukkan salah satu contoh *user story* yang dibuat pada aplikasi manajemen proyek Redmine.



Gambar 3.1 Contoh *User Story*

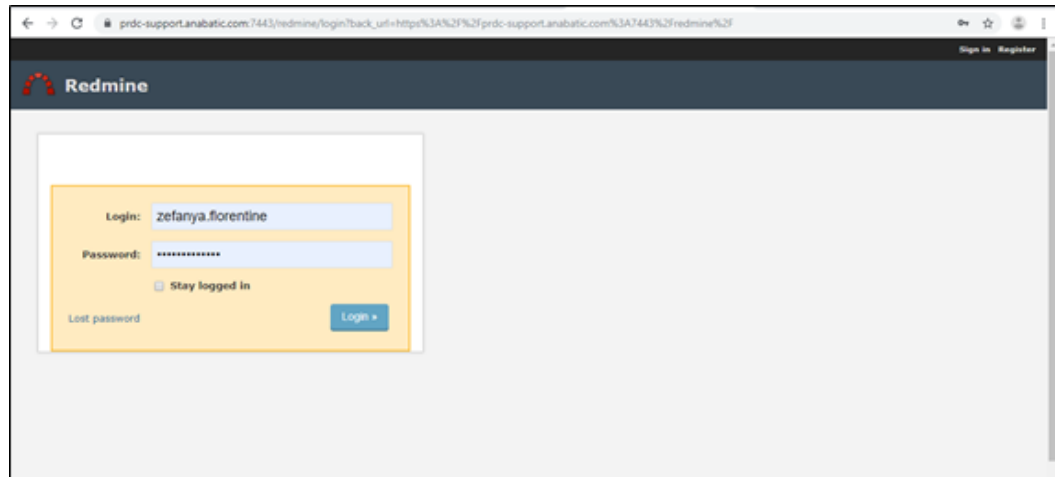
Dalam model *scrum*, proyek dikembangkan dengan serangkaian *sprint*. *Sprint* adalah batasan waktu pengerjaan yang telah ditetapkan dan tidak diperkenankan mengubah konsep ditengah jalan pengerjaan proyek, dimana satu *sprint* berlangsung selama 10 hari kerja dan di akhir *sprint* dilakukan *sprint review* bersama *stakeholder* yang terkait untuk meninjau semua pekerjaan yang telah

diselesaikan dan merencanakan pekerjaan yang akan dilakukan pada *sprint* selanjutnya. Setiap harinya, di dalam *sprint* dilakukan *daily scrum meeting* via aplikasi *Skype* setiap pukul 10.00 dimana setiap anggota tim wajib melaporkan pekerjaan yang akan dilakukan hari ini dan memperbarui status *task* yang akan dikerjakan pada Redmine menjadi “*dev in progress*”. Gambar 3.2 menunjukkan siklus pengerjaan proyek dengan menggunakan *scrum framework*.



Gambar 3.2 Siklus Scrum Framework (Scrum,2018)

Redmine merupakan aplikasi manajemen proyek yang dibuat dengan menggunakan *frameworks Ruby on Rails open source* berbasis *web* untuk manajemen proyek. Redmine mendukung *multiple project* dan memiliki *GanttChart* dan kalender untuk mengolah dokumentasi proyek. Tugas dibagikan kepada *team member* dengan konsep *issue* yang harus dikerjakan dan sistem pelacakan *issue* yang fleksibel (Redmine, 2018). Gambar 3.3 menunjukkan tampilan awal aplikasi Redmine pada PT Anabatic Digital Raya khususnya pada divisi *Product Research and Development*.



Gambar 3.3 Tampilan *Website* Redmine Anabatic Digital Raya

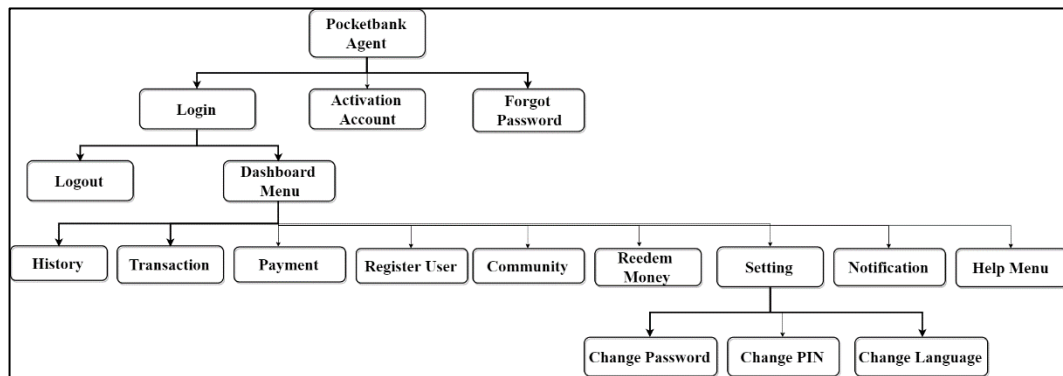
Redmine adalah aplikasi manajemen proyek yang dibuat dengan menggunakan *frameworks Ruby on Rails open source* berbasis *web* untuk manajemen proyek. Redmine mendukung *multiple project* dan memiliki Gantt chart dan kalender untuk mengelola dokumentasi proyek. Tugas dibagikan pada *team member* dengan menggunakan konsep *issue* yang akan dikerjakan dan sistem pelacakan *issue* yang fleksibel (Redmine, 2018). Gambar 3.3 menunjukkan tampilan awal aplikasi *redmine* pada PT Anabatic Digital Raya.

Pelaksanaan kerja magang proyek yang dikembangkan berupa *chatbot*, namun karena terjadi suatu kendala, maka *jobdesk* magang yang diberikan menjadi perancangan aplikasi berbasis *website Pocketbank Agent*. Dalam pelaksanaan kerja magang terbagi menjadi beberapa tahap, yaitu perancangan dan implementasi, dengan penjelasan sebagai berikut.

3.3.1. Perancangan

Tahap perancangan dalam pembuatan aplikasi *Pocketbank Agent* meliputi pembuatan *sitemap*, *data flow diagram*, *flowchart*, dan rancangan antar muka juga merupakan bagian dari tahap perancangan

3.3.1.1. Site Map



Gambar 3.4 *Site Map* Aplikasi Pocketbank Agent.

Gambar 3.4 menunjukkan *site map* Aplikasi Pocketbank Agent, dimana aplikasi ini memiliki beberapa fitur yang terdapat dalam aplikasi *Pocketbank Agent* sebagai berikut.

1. Halaman *Login*

Halaman *Login* adalah halaman yang berguna untuk masuk ke dalam aplikasi. *User* harus mengisi *form username* dan *password* untuk masuk ke dalam menu *dashboard* aplikasi yang terdiri dari berbagai fitur di dalamnya. Dalam menu *login* terdapat fitur simpan *username*, dimana *user* dapat menyimpan *username* dan *link-button forgot password*.

2. Halaman *Activation Account*

Halaman *Activation Account* adalah halaman yang berguna untuk mengaktivasi akun *user* dengan memasukkan nomor telepon *user* dan nama pemilik akun lalu melakukan verifikasi melalau kode *One Time Password* (OTP) yang telah dikirimkan melalui nomor telepon *user*.

3. Halaman *Forgot Password*

Halaman *Forgot Password* adalah halaman yang berguna untuk mengganti

atau mereset *password*, apabila *user* lupa *password*. *Forgot Password* dilakukan dengan memasukkan kembali nomor telepon dan melakukan verifikasi kode *One Time Password* (OTP) lalu *user* dapat mengganti *password* yang baru.

4. Halaman *Dashboard*

Halaman *Dashboard* merupakan halaman inti dari seluruh fitur yang ada di dalam aplikasi *Pocketbank Agent*, dimana pada halaman *dashboard*, *user* dapat melihat *summary* dari seluruh kegiatan yang pernah dilakukan terhadap aplikasi seperti *transaction history*, antrian, jumlah *agent*, data statistika transaksi, jumlah saldo, jumlah uang kas, anggota komunitas, informasi akun.

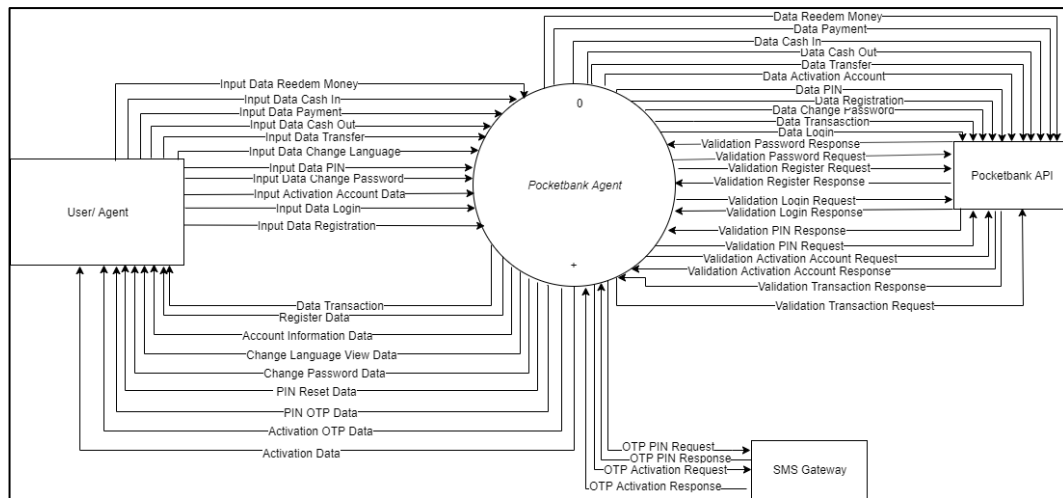
3.3.1.2. Data Flow Diagram (DFD)

Data *Flow Diagram* menggambarkan aliran data yang mengalir pada Aplikasi *Pocketbank Agent* yang sudah dibuat.

A. Diagram Konteks

Gambar 3.5 menunjukkan Data *Flow Diagram* Konteks dari Aplikasi *Pocketbank Agent*. Pada sistem ini terdapat 3 entitas, yaitu *User* atau *Agent*, *Pocketbank API*, dan *SMS Gateway*. Dimana dalam aplikasi *user* dapat melakukan login, mengaktivasi akun, melakukan registrasi *member*, melakukan transaksi, melakukan *setting* seperti mengganti *password*, mengganti PIN transaksi, mengganti bahasa, sedangkan *Pocketbank API* harus dapat melakukan validasi, autentifikasi, otorisasi, dan menyimpan semua transaksi *user* dalam sebuah *database* yang diletakkan pada sebuah *server* dan menggunakan *Application Programming Interface* (API) untuk berkomunikasi antar *user* dan aplikasi. Dalam beberapa fitur pada Aplikasi *Pocketbank Agent* harus menggunakan *SMS Gateway*

berupa *One Time Password* validasi seperti pada Fitur *Forgot Password*, *Forgot PIN*, *Activation Account*.

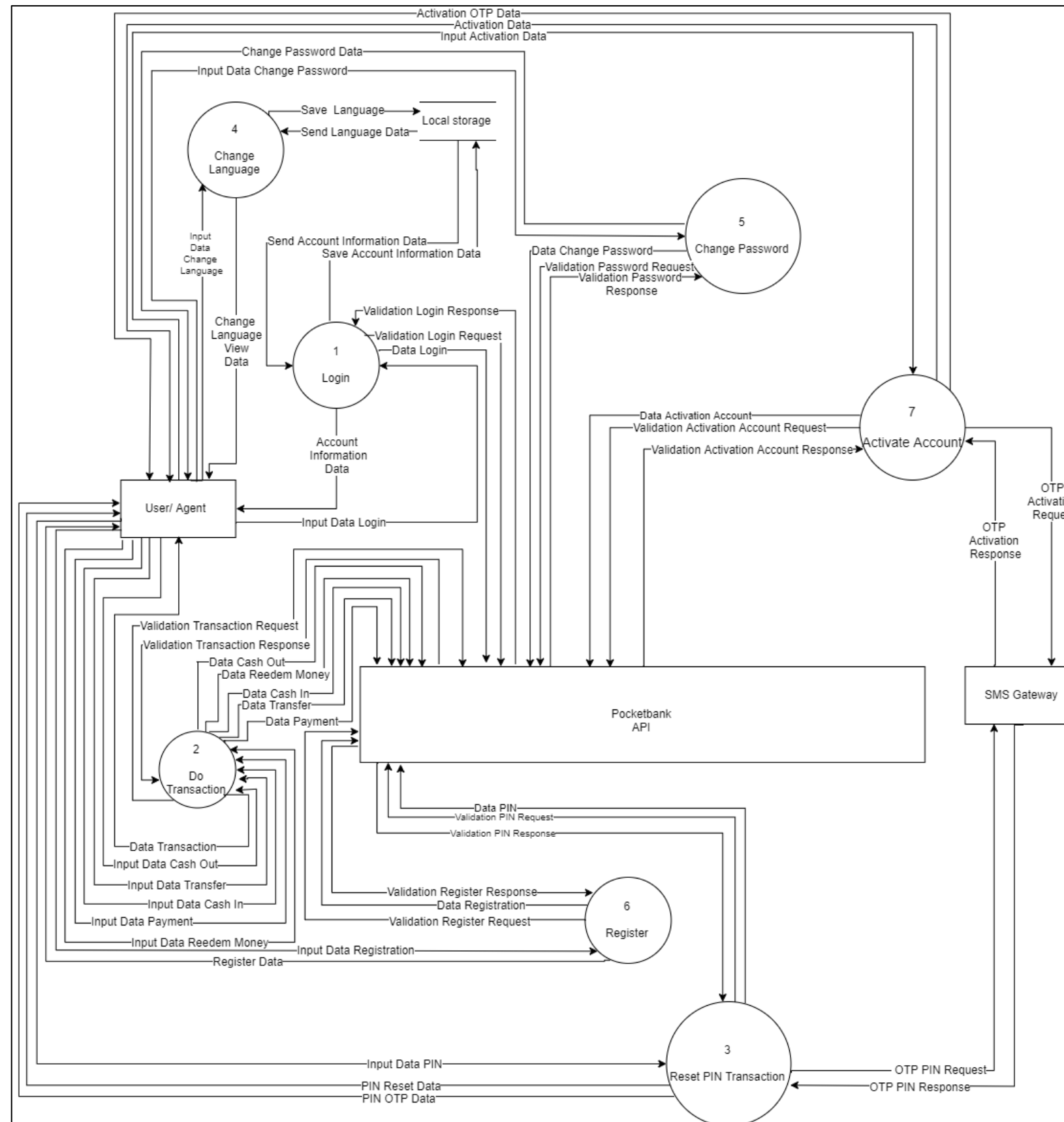


Gambar 3.5 Data Flow Diagram Konteks Aplikasi Pocketbank Agent

B. Data Flow Diagram Level 1

Gambar 3.6 menunjukkan Data Flow Diagram Level 1 dimana aliran data digambarkan secara rinci. Pada Gambar 3.6 menunjukkan proses yaitu *login*, *do transaction*, *Reset PIN Transaction*, *Change Password*, *Change Language*, *Activate Account*, dan *Register*. Ketika *user* ingin melakukan proses *login*, maka *user* harus memasukkan data *login* berupa *input username* dan *password*, lalu sistem akan memeriksa *login validation* untuk memastikan bahwa *data login* yang dimasukkan *user* terdaftar pada *database* yang diwakilkan dengan *API*, setelah selesai *login*, maka data *user* akan dimasukkan kedalam *local storage* yang bertujuan agar *user* tidak melakukan *login* berulang kali ketika *reload* halaman.

User juga dapat melakukan kegiatan transaksi seperti setor tunai, tarik tunai, transfer, pembelian *m-commerce*, minta uang dan kirim uang kepada sesama komunitas, dan melakukan pencairan uang bonus dengan memasukkan *data transaction* sesuai dengan jenis transaksi yang diinginkan.



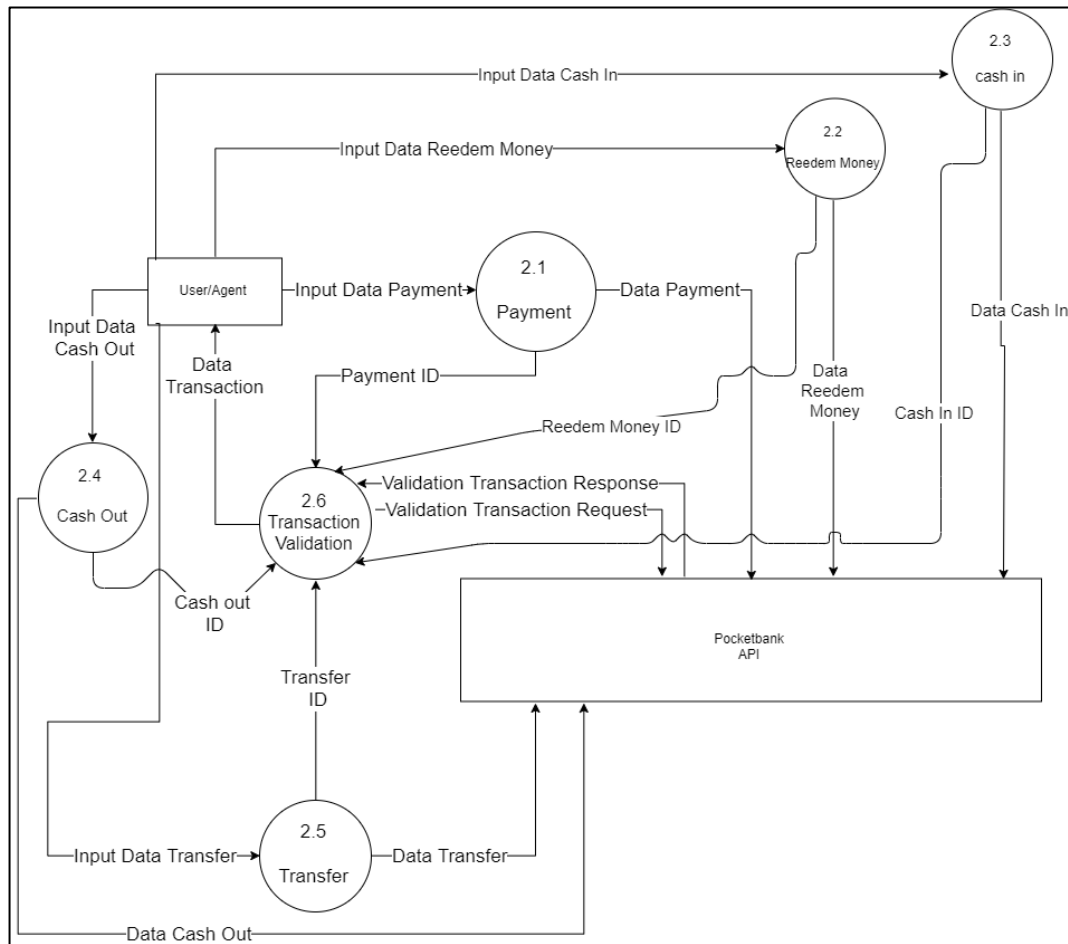
Gambar 3.6 Data Flow Diagram Level

C. Data Flow Diagram Level 2 (Transaction)

Gambar 3.7 menggambarkan Data Flow Diagram yang terjadi dalam proses *transaction*. Dapat dilihat dalam Data Flow Diagram terdapat lima proses transaksi yang dapat dilakukan oleh *user*, yaitu *cash out*, *cash in*, *payment*, *reedem money*, dan transfer. Proses *cash out* merupakan proses tarik tunai, dimana *user* dapat melakukan tarik tunai sesuai dengan nominal yang diinginkan dengan memasukkan data *cash out* seperti nominal dan nomor akun lalu *database* akan melakukan validasi tentang transaksi tarik tunai yang terjadi. Proses *paymen* dapat dilakukan *user* jika *user* ingin membeli *product m-commerce* misalnya seperti *top up* pulsa, maka *user* dapat mengisi *form data payment* dan sistem akan melakukan konfirmasi sehingga *user* akan diminta untuk memasukkan PIN transaksi yang menandakan terjadinya transaksi sukses terhadap pembelian *product m-commerce*.

User juga dapat melakukan setor tunai. Pada proses *Cash In*, *user* diminta untuk memasukkan data *cash in* yang berupa nomor akun dan nominal yang ingin dimasukkan, lalu *system* akan melakukan validasi dengan meminta *user* untuk melakukan konfirmasi terjadinya proses *cash in* dengan memasukkan PIN transaksi yang menandakan bahwa proses setor tunai telah berhasil dilakukan. Proses transfer terjadi jika *user* ingin melakukan transfer uang kepada orang lain, maka dalam proses transfer, *user* diminta untuk memasukkan data transfer berupa nomor rekening dan nomor rekening tujuan beserta nominalnya, lalu *system* akan melakukan validasi dengan meminta *user* memasukkan PIN transaksi sebagai tanda *user* setuju melakukan transaksi transfer pada waktu tertentu. Proses *reedem money* dilakukan ketika *user* hendak melakukan pencairan uang yang terdapat pada aplikasi, maka *user* akan diminta untuk memasukkan data berupa nomor akun dan

nominal yang ingin dicairkan, maka *system* akan meminta *database* pada Pocketbank API untuk melakukan pencairan uang sesuai dengan nominal yang diinginkan oleh *user*.



Gambar 3.7 Data Flow Diagram Level 2 *Transaction*

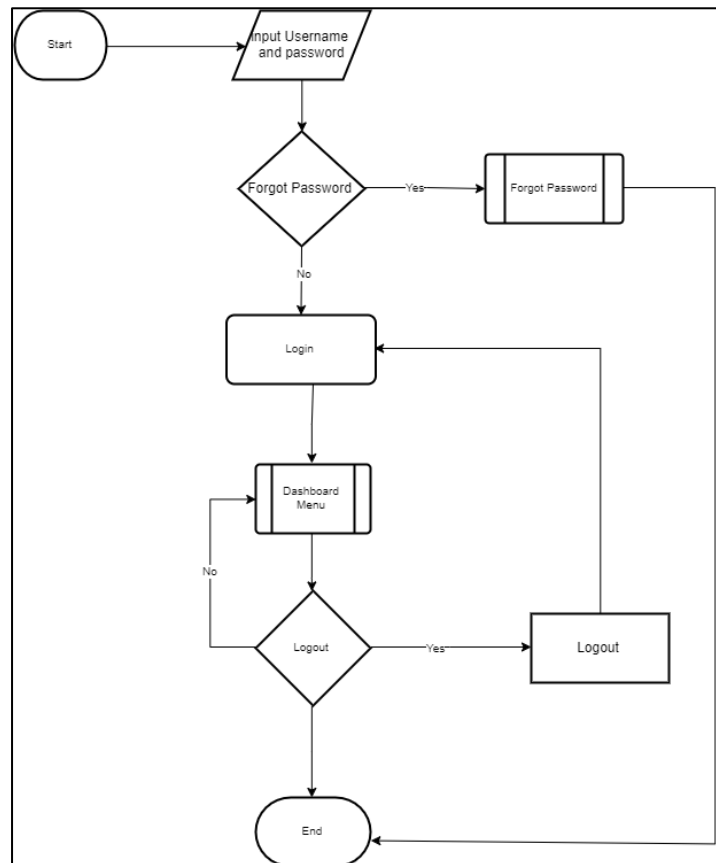
3.3.1.3. Flowchart

Flowchart merupakan diagram yang menunjukkan alur kerja program.

A. Flowchart Aplikasi Pocketbank Agent

Gambar 3.8 menunjukkan *flowchart* Aplikasi Pocketbank Agent, dimana *flowchart* tersebut adalah inti dari penggunaan Aplikasi Pocketbank Agent. Dimana dalam menggunakan Aplikasi Pocketbank Agent, *user* diminta untuk melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password*, bila *user* lupa

password akun, maka *user* dapat menekan *button Forgot Password*. Setelah selesai *login*, maka *user* akan masuk ke halaman *dashboard* pada Aplikasi Pocketbank Agent. Jika *user* hendak keluar dari aplikasi, maka *user* dapat menekan *button logout*. Maka *user* akan dikembalikan ke halaman *login*.

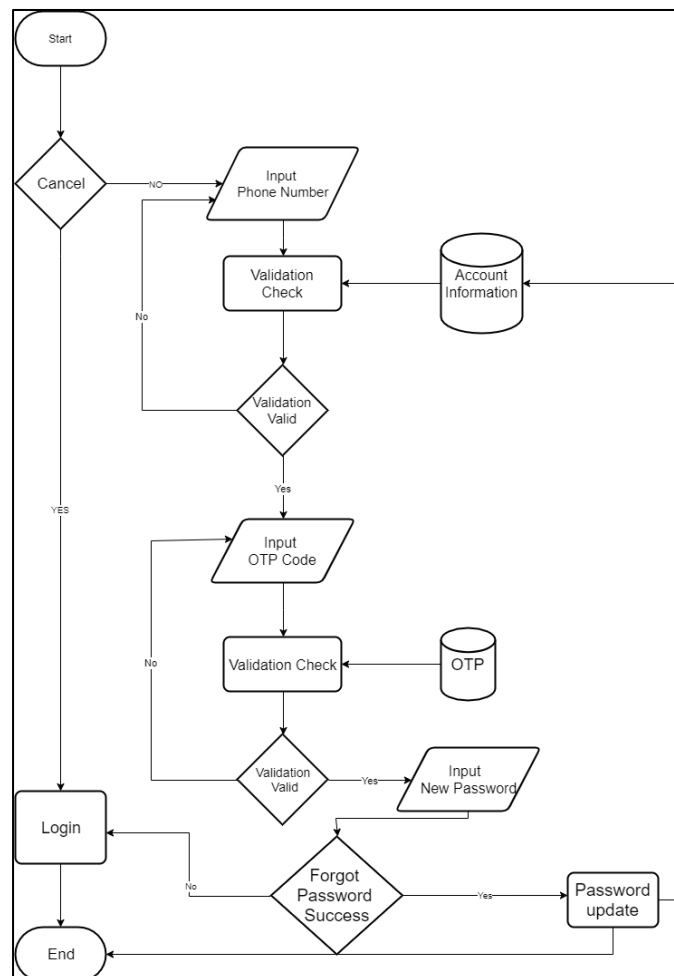


Gambar 3.8 Flowchart Aplikasi Pocketbank Agent

B. Flowchart Forgot Password

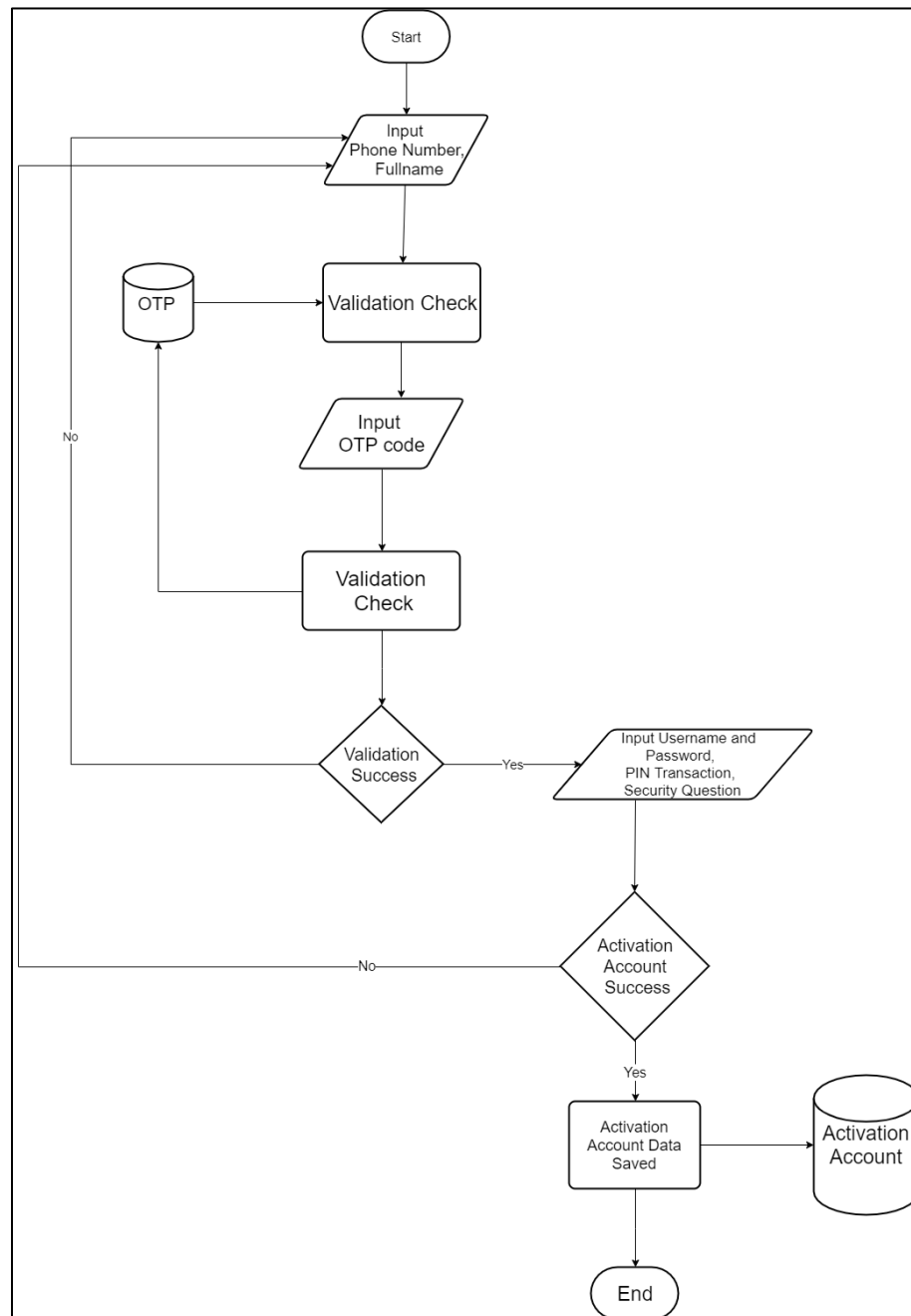
Gambar 3.9 menggambarkan *flowchart Forgot Password* dimana *user* harus memasukkan nomor ponsel yang terdaftar pada Aplikasi Pocketbank Agent lalu memasukan kode OTP yang dikirimkan ke nomor ponsel yang dimasukkan sebelumnya. Sebelum mengirimkan kode OTP yang akan dikirimkan ke nomor ponsel yang diinput, *system* melakukan validasi terlebih dahulu pada nomor telepon

yang diinput, jika nomor telepon tersebut *valid*, maka *system* akan mengirimkan kode OTP ke nomor telepon *user* yang diinput sebelumnya. Setelah *user* memasukkan kode OTP tersebut, maka *system* akan melakukan validasi kembali akan kode OTP yang dimasukkan oleh *user*, jika kode tersebut *valid*, maka *user* dapat mereset *password* dengan *password* yang baru dan secara otomatis tampilan ke menu *login* dan mengeluarkan *notification alert* yang menandakan kalau proses *forgot password* sukses dan *user* diminta untuk *login* dengan *password* yang baru. Jika proses *forgot password cancel* maka akan dibalikan ke menu *login* dan *database* pada *system* secara otomatis melakukan *update* terhadap *password* yang baru.



Gambar 3.9 Flowchart Forgot Password

C. Flowchart Activation Account

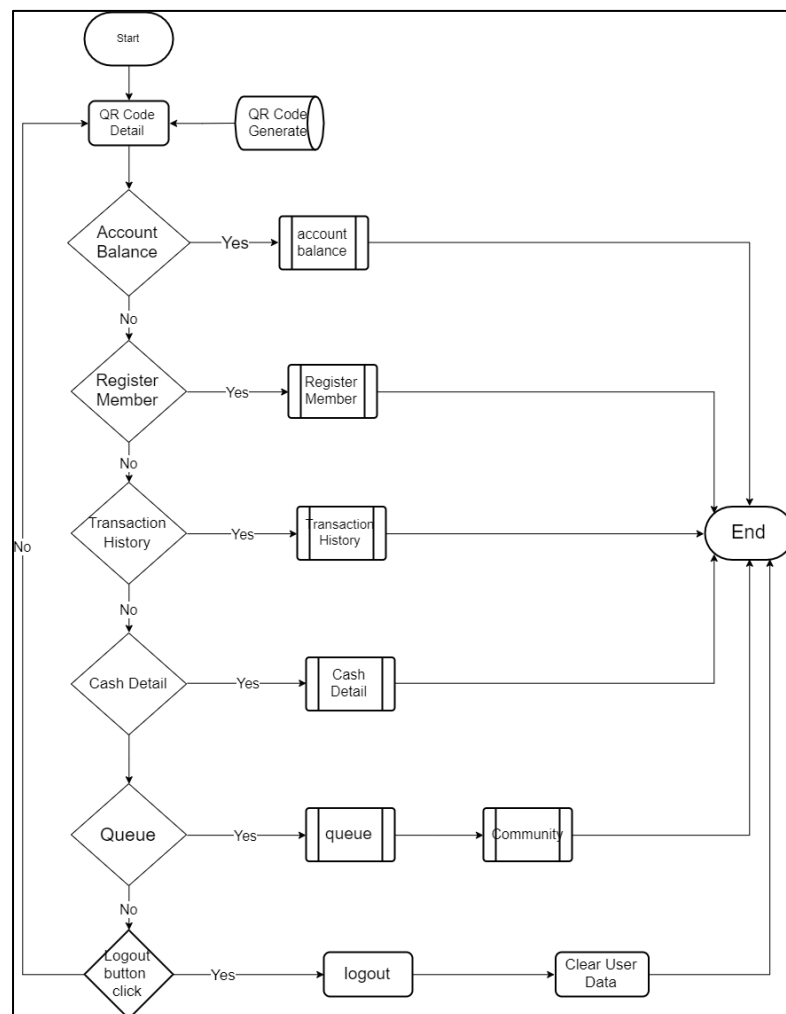


Gambar 3.10 Flowchart Activation Account

Gambar 3.10 menggambarkan *flowchart Activation Account* dimana pertama *user* harus memasukkan nomor ponsel dan nama lengkap lalu menginput kode OTP yang dikirimkan ke nomor ponsel yang didaftarkan setelah itu *user* baru dapat mengaktifkan *account* dengan membuat *username*, *password*, dan

transaction PIN, lalu memilih salah satu dan menjawab *security question* yang diberikan aplikasi. Setelah selesai, *user* dapat menggunakan *account* yang telah didaftarkan. Dalam Aplikasi Pocketbank Agent Fitur *Activation Account* dibuat secara terpisah dari fitur lainnya.

D. Flowchart Dashboard



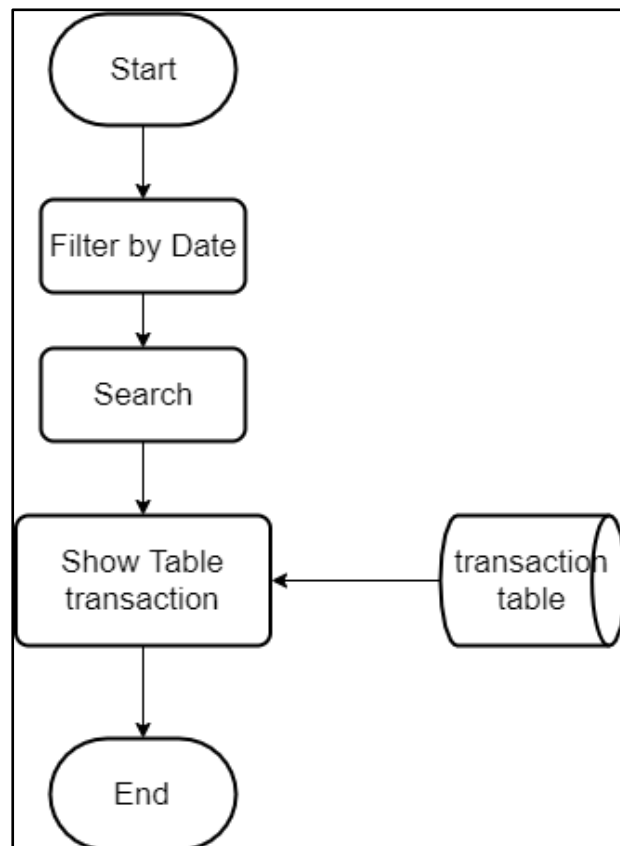
Gambar 3.11 Flowchart Dashboard

Gambar 3.11 menggambarkan *flowchart* halaman *dashboard* dimana dalam Menu *Dashboard* merupakan *summary* atau inti dari semua keterangan yang pernah dilakukan *user* terhadap aplikasi seperti melihat *trasaction history*, jumlah *member* yang pernah didaftarkan, *community*, saldo akun, *cash*, Dalam Menu *Dashboard*,

user juga dapat melihat *detail username* yang berisi *QR Code* yang sudah di-*generate* berdasarkan *QR Code Standar Indonesia*.

E. Flowchart Account Balance

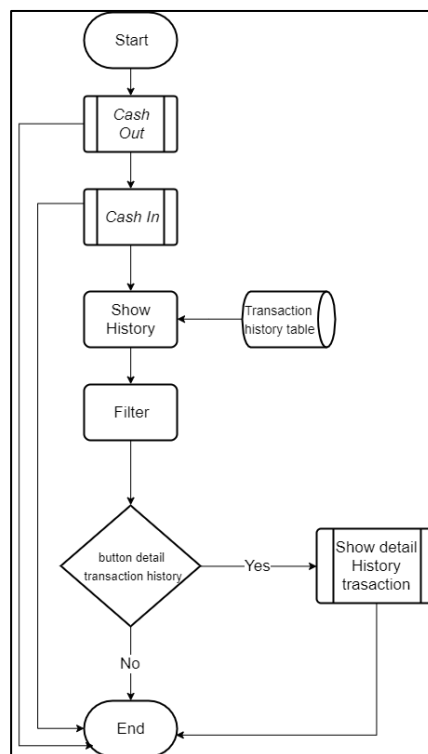
Gambar 3.12 menggambarkan *flowchart* bagian *account balance* dimana *user* dapat melihat saldo akun beserta *detail* transaksi yang pernah terjadi pada waktu tertentu. Data *detail* transaksi sudah disediakan dalam bentuk tabel yang dinamakan *transaction table* yang secara otomatis di-*update* ketika terjadi aktivitas transaksi. Pada bagian *account balance*, *user* juga dapat melakukan *search* untuk mencari transaksi dan *filter by date* untuk mencari transaksi yang terjadi pada *range* tanggal tertentu. Fitur *search* dan *filter* dapat digunakan untuk mempermudah pencarian transaksi berdasarkan nama dan waktu tertentu.



Gambar 3.12 Flowchart Account Balance

F. Flowchart Cash Detail

Gambar 3.13 menggambarkan *flowchart cash detail* dimana dalam fitur ini, *user* dapat melihat nominal uang kas yang dapat dicairkan, *user* juga dapat melakukan tarik tunai maupun setor tunai. Dalam *cash detail*, *user* juga dapat melihat seluruh transaksi yang pernah terjadi pada waktu tertentu yang sudah disediakan dalam *table transaction history detail*. *User* juga dapat melakukan *filter by date* berdasarkan *range* waktu tertentu untuk mempermudah pencarian transaksi yang pernah dilakukan sebelumnya.

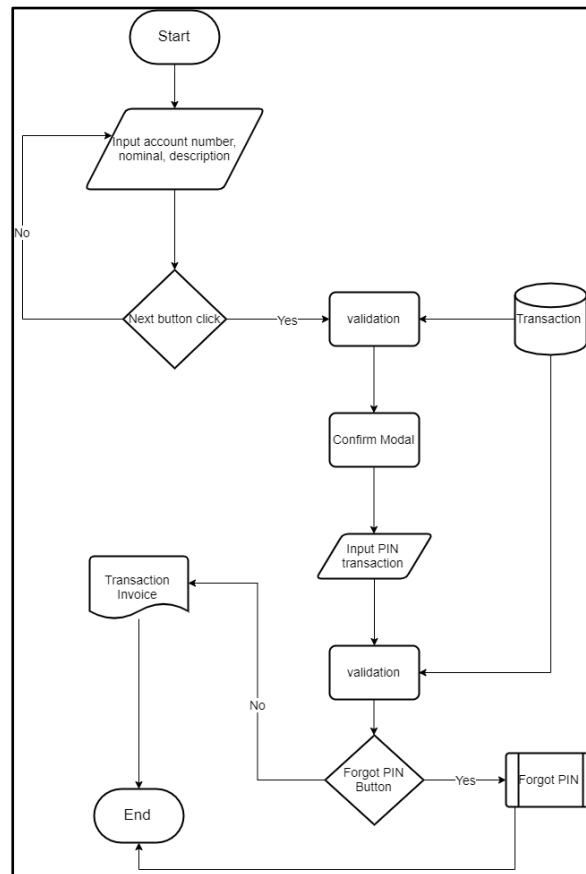


Gambar 3.13 Flowchart Cash Detail

G. Flowchart Cash In

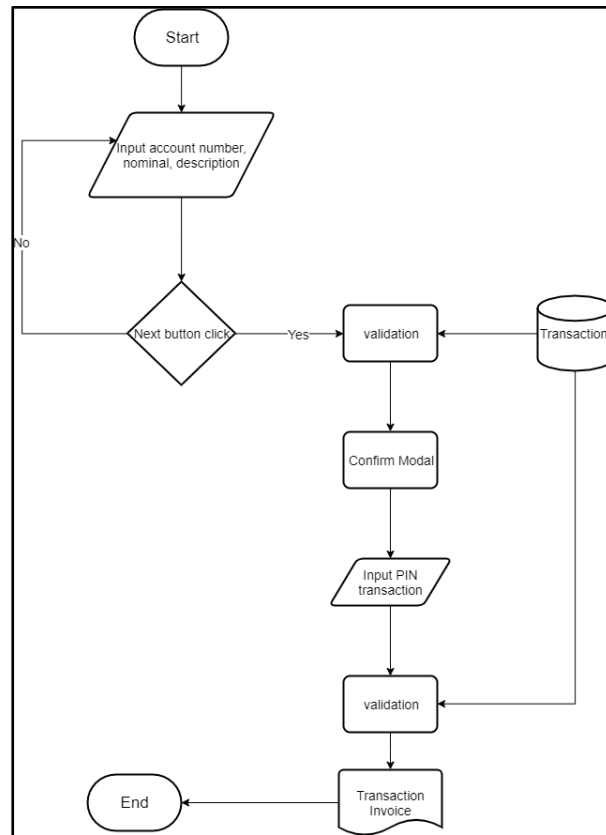
Gambar 3.14 menunjukkan Flowchart Cash In, dimana dalam fitur ini *user* dapat melakukan setor tunai. Fitur Cash In terdapat pada Halaman Dashboard, halaman *transaction history*, dan halaman *transaction*. Jika *user* melakukan setor tunai pada halaman *dashboard* dan *transaction history*, maka *user* tidak perlu

memasukkan nomor akun, sedangkan jika *user* melakukan pada fitur halaman *transaction* sendiri *user* dapat melakukan setor tunai dengan memasukkan nomor akun dan nominal yang diinginkan dan jika diperlukan, *user* dapat menambahkan deskripsi transaksi. Setelah itu akan keluar *modal receipt* yang menandakan *user* ingin melakukan setor tunai pada saat itu, lalu *user* diminta memasukkan PIN transaksi untuk menyetujui terjadinya transaksi. Jika *user* lupa PIN, maka *user* dapat menekan *Button Forgot PIN*, lalu *user* akan diarahkan pada halaman *Forgot PIN*. Setelah PIN sudah dimasukkan, maka Aplikasi akan memberikan bukti transaksi sukses dan *user* dapat melakukan *print invoice*.



Gambar 3.14 Flowchart Cash In

H. Flowchart Cash Out



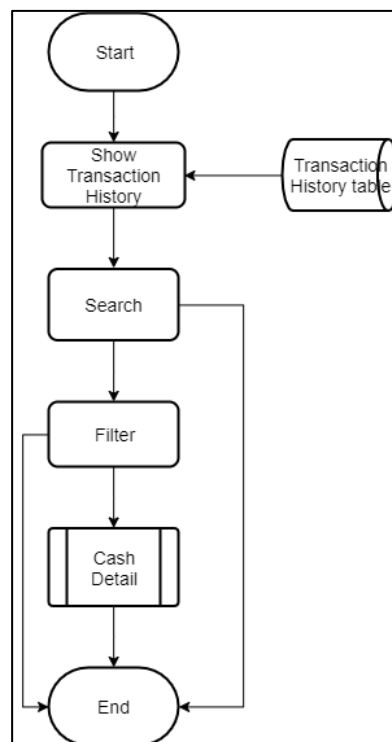
Gambar 3.15 Flowchart Cash Out

Gambar 3.15 menggambarkan *flowchart cash out* dimana *user* dapat melakukan tarik tunai dengan nominal yang diinginkan. Fitur *cash out* ada pada halaman *dashboard*, halaman *history*, halaman *transaction*. Jika *user* menggunakan fitur *cash out* pada halaman *dashboard* dan halaman *history*, *user* tidak diminta untuk memasukkan nomor akun, *user* cukup memasukkan nominal dan menambahkan *description* bila diperlukan, maka *user* akan diminta untuk melakukan konfirmasi lalu memasukkan PIN transaksi. Bila *user* lupa PIN transaksi maka *user* dapat menekan *button Forgot PIN* lalu *user* akan diarahkan pada halaman *Forgot PIN*. Jika *user* melakukan tarik tunai pada Fitur *cash out* yang ada pada halaman *transaction*, maka *user* perlu memasukkan nomor akun dan nominal yang ingin diambil, bila perlu *user* dapat menambahkan *transaction description*. Setelah itu,

user diminta untuk melakukan konfirmasi dan memasukkan PIN transaksi. Setelah semua proses selesai dilalui, maka *user* akan diberikan *invoice* yang dapat dicetak sebagai bukti fisik terjadinya transaksi tarik tunai.

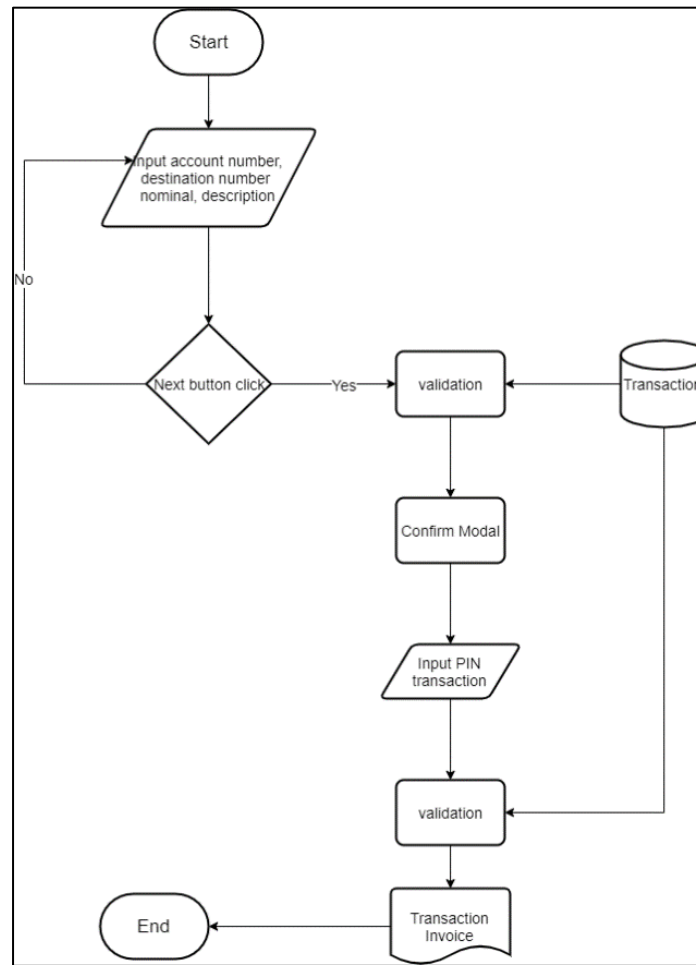
I. Flowchat History

Gambar 3.16 menunjukkan *flowchart* halaman *history*. Dimana pada halaman ini, *user* dapat melihat seluruh transaksi yang pernah dilakukan oleh *user*. *User* diperkenankan untuk melakukan *filter by date* dan *search* untuk mempermudah pencarian transaksi. *User* juga dapat melihat *cash detail* serta riwayat penarikan dan setor tunai yang pernah dilakukan pada waktu tertentu.



Gambar 3.16 Flowchart History

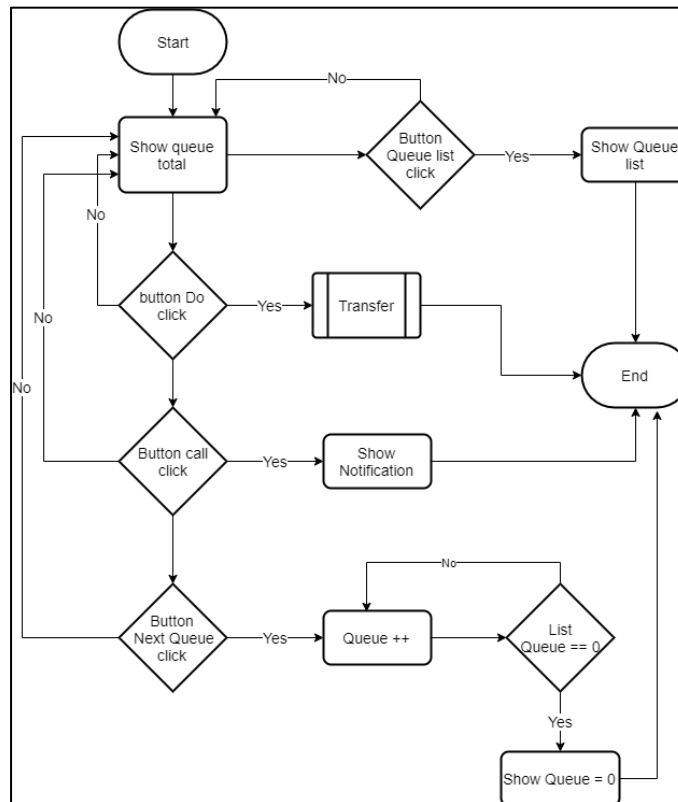
J. Flowchart Transfer



Gambar 3.17 Flowchart Transfer

Gambar 3.17 menggambarkan *flowchart transfer* yang ada pada menu *transaction*. Dimana *user* diminta untuk memasukkan nomor akun dan nomor akun tujuan, memasukkan nominal yang hendak ditransfer dan menambahkan deskripsi bila perlu, lalu *user* akan diminta untuk melakukan konfirmasi dengan memasukkan PIN transaksi, dan secara otomatis *system* akan melakukan validasi pada *database* terhadap PIN yang dimasukkan oleh *user* setelah selesai maka *user* akan diberikan bukti transaksi yang dapat dicetak sebagai bukti kalau proses transfer telah sukses dilakukan.

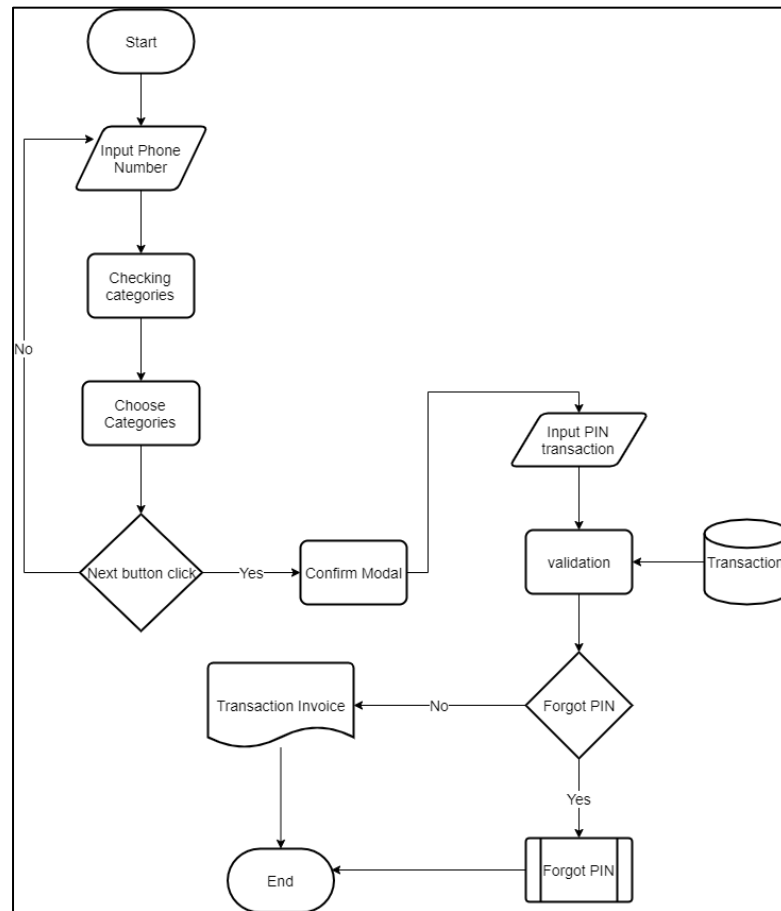
K. Flowchart Queue



Gambar 3.18 Flowchart Queue

Gambar 3.18 menggambarkan *flowchart queue* dimana *user* dapat melihat jumlah antrian dan nama *member* yang sedang melakukan antri. Dalam Aplikasi Pocketbank Agent jumlah antrian digunakan untuk melakukan transaksi berupa *transfer* kepada *member* lain. *User* dapat menekan *button call* untuk memberitahu bahwa proses transaksi sedang dilakukan sehingga *member* lain dapat memperoleh *notification*. Bila suatu transaksis sudah selesai dikerjakan, maka *user* dapat menekan *button next* agar *member* selanjutnya dapat diproses.

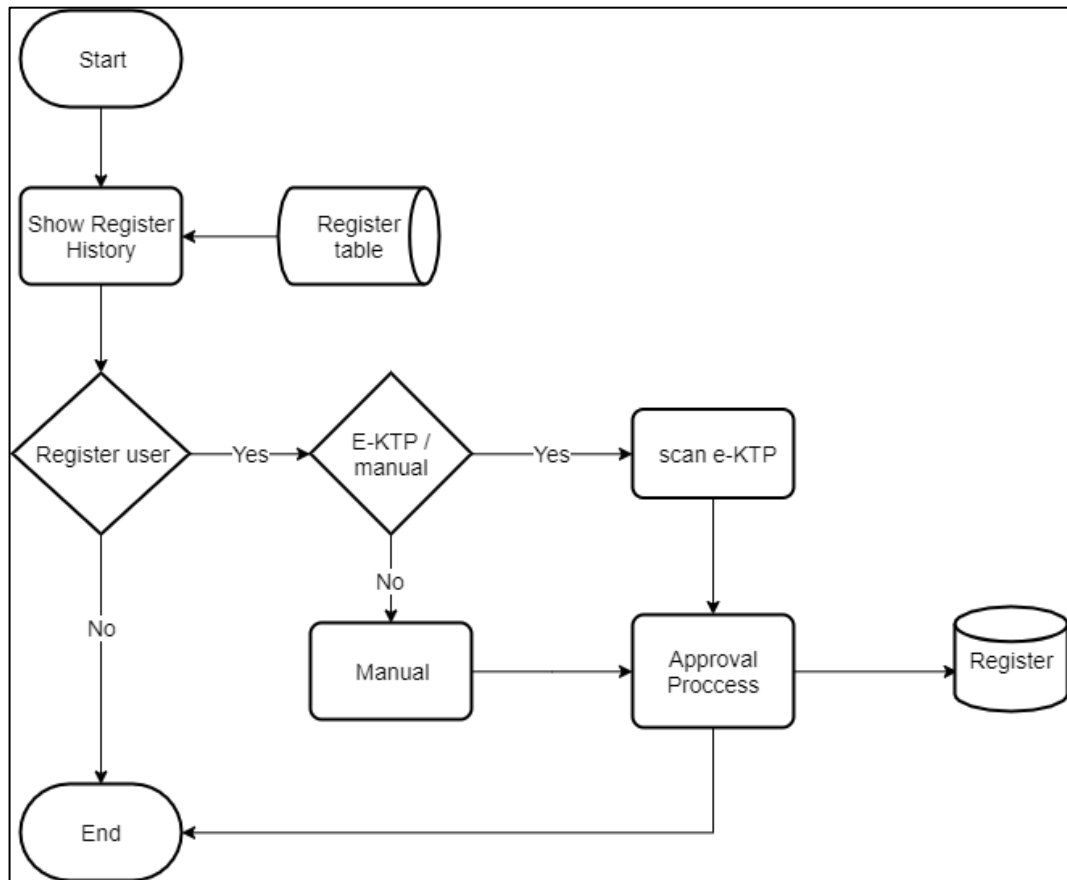
L. Flowchart Payment



Gambar 3.19 Flowchart Payment

Gambar 3.19 menggambarkan *flowchart payment* pada salah satu *m-commerce* dimana *flowchart* tersebut menunjukkan alur untuk pembelian pulsa. Pada fitur pembelian pulsa, *user* diminta untuk memasukkan nomor ponsel yang hendak di-*top up*, sistem aplikasi mendeteksi secara otomatis kategori nomor telepon sesuai dengan operatornya, lalu aplikasi akan memunculkan beberapa kategori beserta harga dari operator tersebut yang dapat *user* pilih sesuai dengan kebutuhan, setelah memilih, *user* diminta untuk konfirmasi dengan memasukkan PIN transaksi, apabila *user* lupa akan PIN transaksi maka *user* akan diarahkan kedalam menu *Forgot PIN*. Jika pembelian *m-commerce* sukses, maka *user* akan diberikan *invoice* yang dapat dicetak.

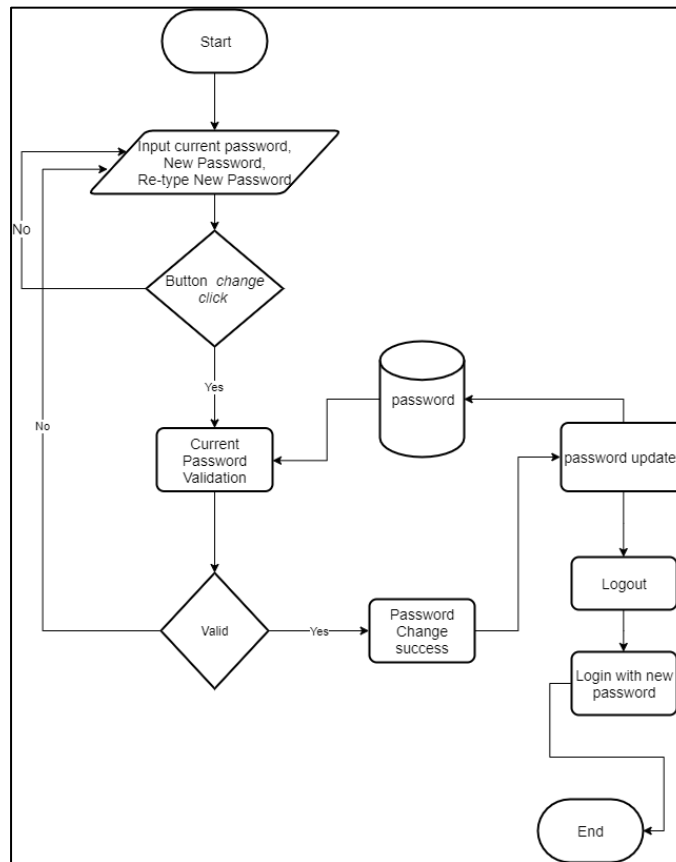
M. Flowchart Register



Gambar 3.20 Flowchart Register

Gambar 3.20 menggambarkan *flowchart register*, dimana *user* dapat melihat *list member* yang pernah *user* daftarkan beserta status registrasinya yang disediakan dalam bentuk tabel register. Bila *user* menekan *button register* maka *user* dapat memilih untuk mendaftarkan *member*-nya dengan *manual registration* atau dengan melakukan *scan e-ktp*. Bila *user* memilih mendaftarkan *manual*, maka *user* diminta untuk mengisi *form*. Setelah seluruh proses dilalui, maka sistem akan memberi *notification* untuk proses *approval*.

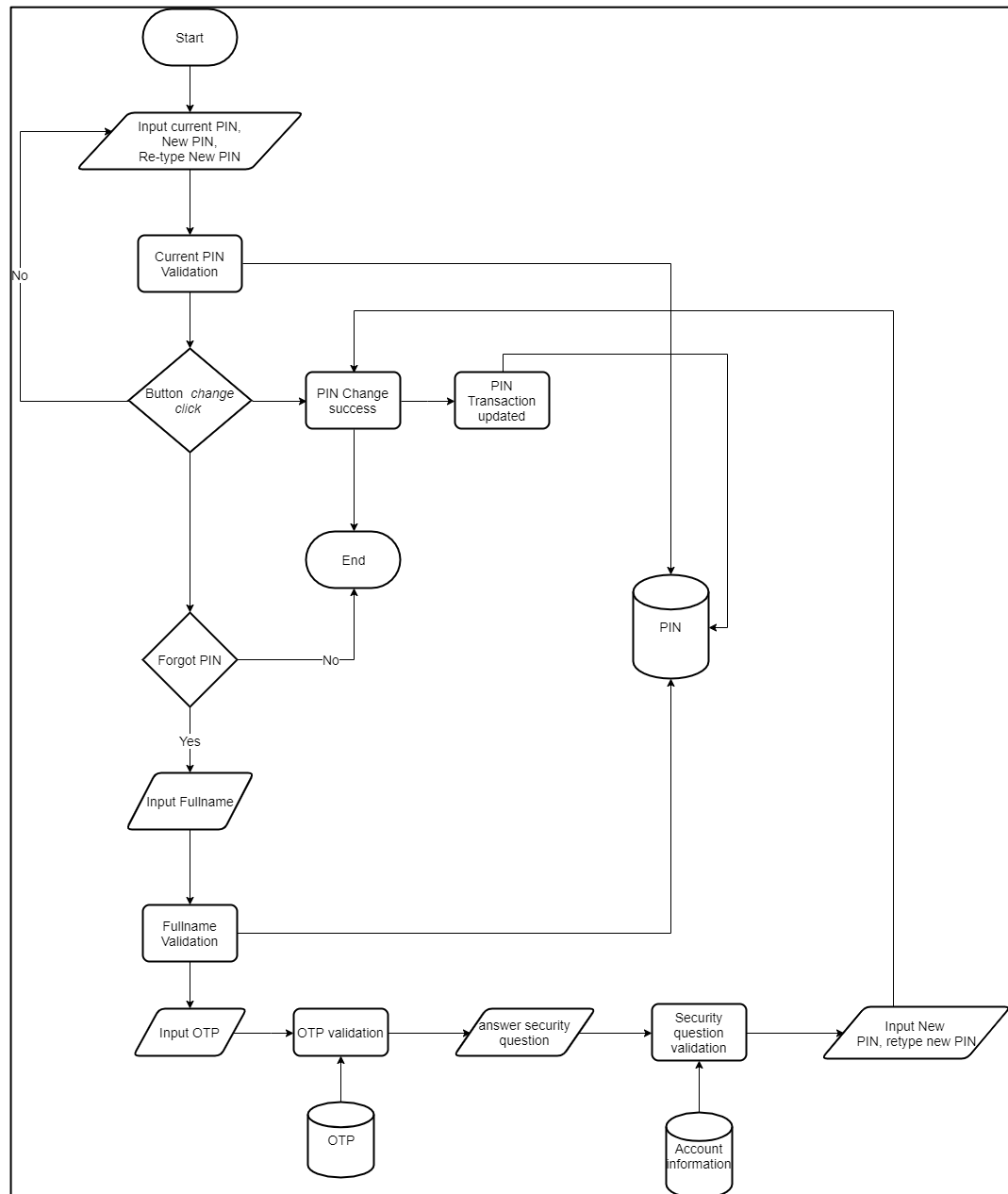
N. Flowchart Change Password



Gambar 3.21 Flowchart Change Password

Gambar 3.21 menggambarkan *flowchart* mengubah password, dimana *user* harus menginput *current password*, *new password*, dan *re-type new password*. Secara otomatis, aplikasi akan mendeteksi validasi terhadap *current password*. Jika *change password* sukses, maka aplikasi akan ter-*logout* secara otomatis dan *user* diminta untuk *login* menggunakan *password* yang baru.

O. Flowchart Forgot PIN

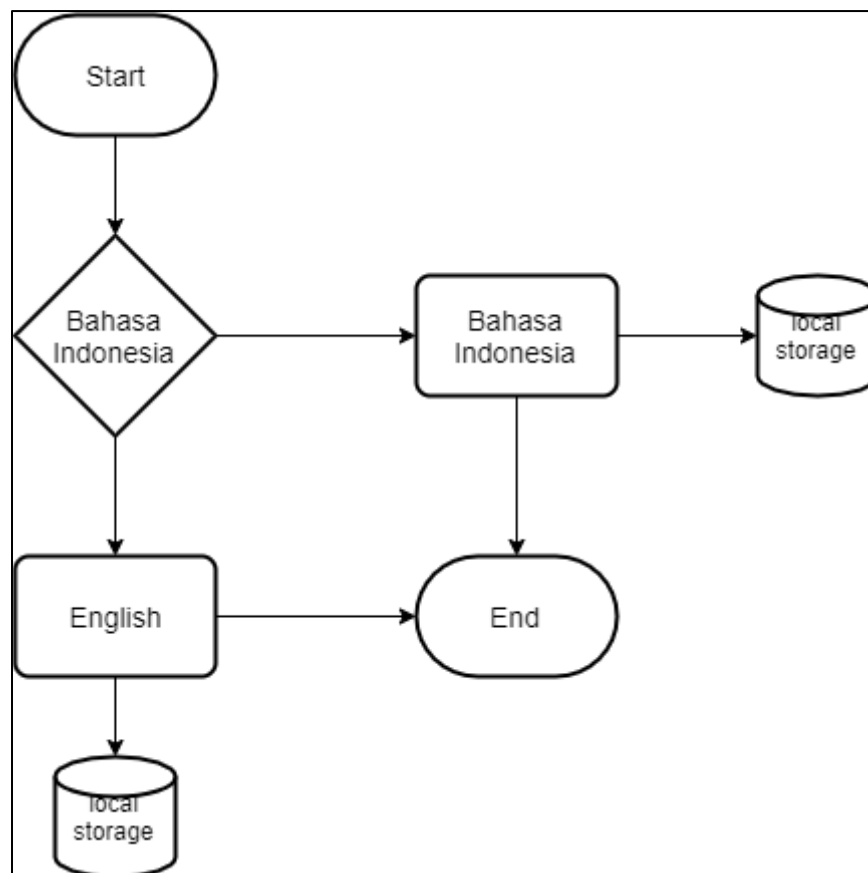


Gambar 3.22 Flowchart Forgot PIN

Gambar 3.22 menggambarkan *flowchart forgot PIN*, dimana pada fitur ini, *user* dapat mengganti PIN transaksi dengan memasukkan *current PIN*, *new PIN*, dan *re-type new PIN*. Secara otomatis aplikasi akan melakukan validasi untuk mengganti PIN. Bila *user* lupa PIN transaksi, *user* dapat menekan *button forgot PIN*, lalu *user* diminta untuk memasukkan nama lengkap sesuai dengan yang

terdaftar pada aplikasi, lalu sistem akan mengirimkan kode OTP pada nomor telepon *user* yang terdaftar pada aplikasi dan meminta *user* mengetikkan kembali kode tersebut, lalu *user* akan diberi pertanyaan berupa *security question* yang pernah *user* jawab sebelumnya ketika mendaftarkan *account*. Setelah itu *user* dapat membuat PIN transaksi baru.

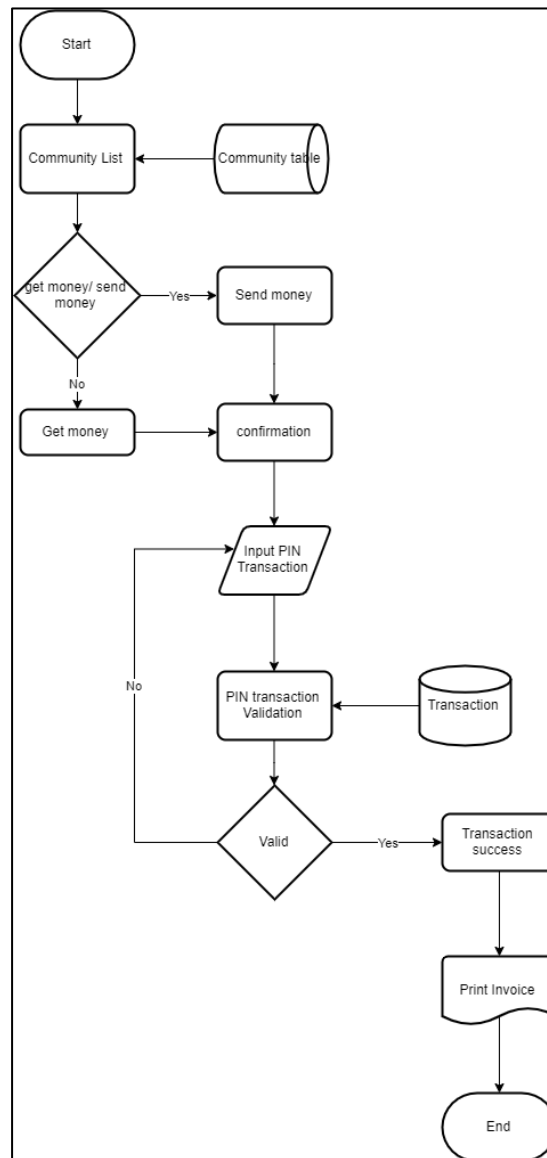
P. Flowchart Change Language



Gambar 3.23 Flowchart Change Language

Gambar 3.23 menggambarkan *flowchart change language*, dimana pada fitur ini terdapat dua bahasa yang dapat dipilih oleh *user* yaitu Bahasa Indonesia dan *English*. Bahasa yang dipilih akan disimpan ke dalam *local storage* agar bahasa tidak kembali terganti ke bahasa sebelumnya atau bahasa *default* Aplikasi Pocketbank Agent ketika melakukan proses *reload* pada halaman *website*.

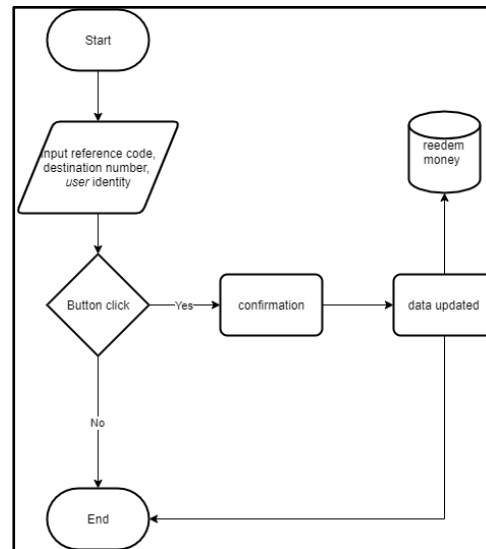
Q. Flowchart Community



Gambar 3.24 Flowchart Community

Gambar 3.24 menggambarkan *flowchart community*, dimana *user* dapat melihat komunitas yang *user* buat yang dapat digunakan untuk meminta dan mengirim uang kepada sesama komunitas. *Community* secara otomatis dapat dilihat pada *table* yang sudah disediakan oleh *table community* yang didapat ketika *user* mendaftarkan *member* lain pada Fitur Register.

R. *Flowchart Reedem Money*



Gambar 3.25 Flowchart Reedem Money

Gambar 3.25 menggambarkan *flowchart reedeem money*, dimana *user* dapat mencairkan uangnya dengan memasukan *reference code*, *account number*, dan nama penerima yang akan menerima uangnya. Setelah itu *user* akan mendapat konfirmasi bila uangnya selesai dicairkan.

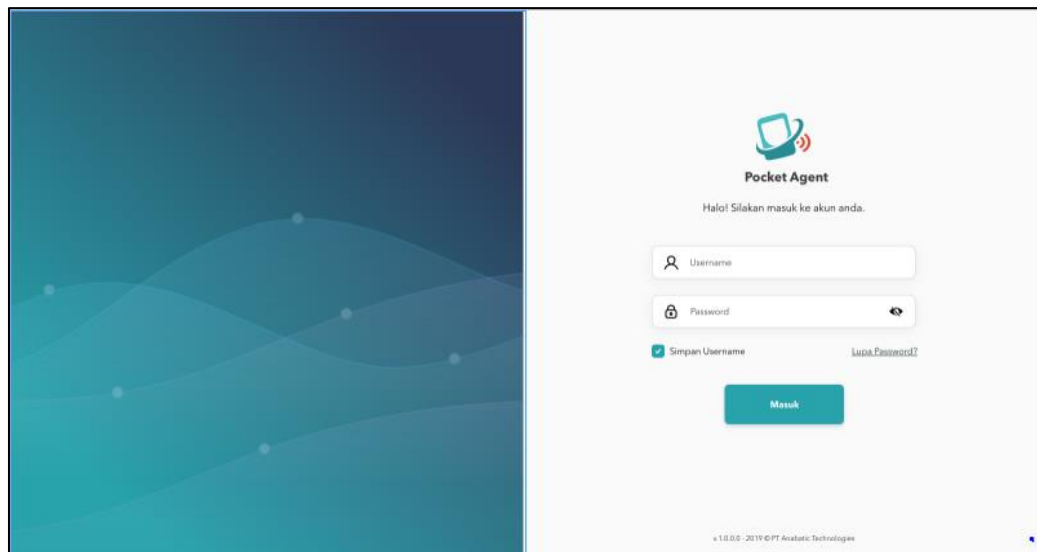
3.3.1.4. Rancangan Antarmuka

Rancangan antarmuka dibuat dengan menggunakan *Adobe Illustrator* dan menggunakan aplikasi *zeplin* dan *invision* untuk lebih menjelaskan tampilan *website* secara keseluruhan.

1. Halaman Mock Up Login

Gambar 3.26 menggambarkan rancangan halaman *login* aplikasi Pocketbank Agent berbasis *website*. Pada halaman *login* terdapat logo Aplikasi Pocketbank Agent dan untuk melakukan proses *login*, *user* perlu memasukkan *username* dan *password* untuk *login*. Pada *login*, terdapat *checkbox* Simpan *Username* yang dapat digunakan oleh *user* untuk menyimpan *username* agar ketika melakukan *logout* atau melakukan *reload* halaman *user* tidak perlu mengetikkan

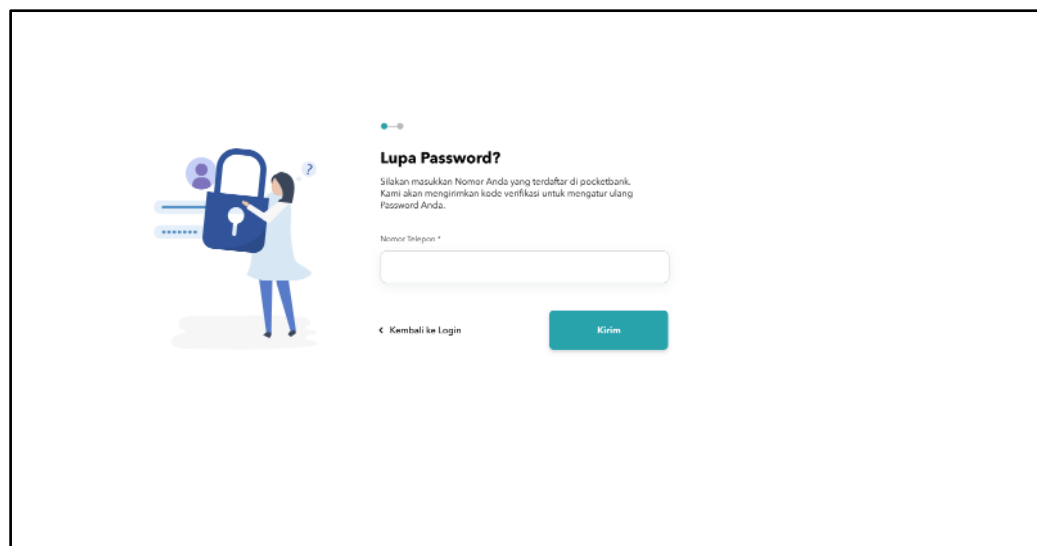
username kembali.



Gambar 3.26 Halaman Mock Up Login

2. Halaman *Mock Up Forgot Password*

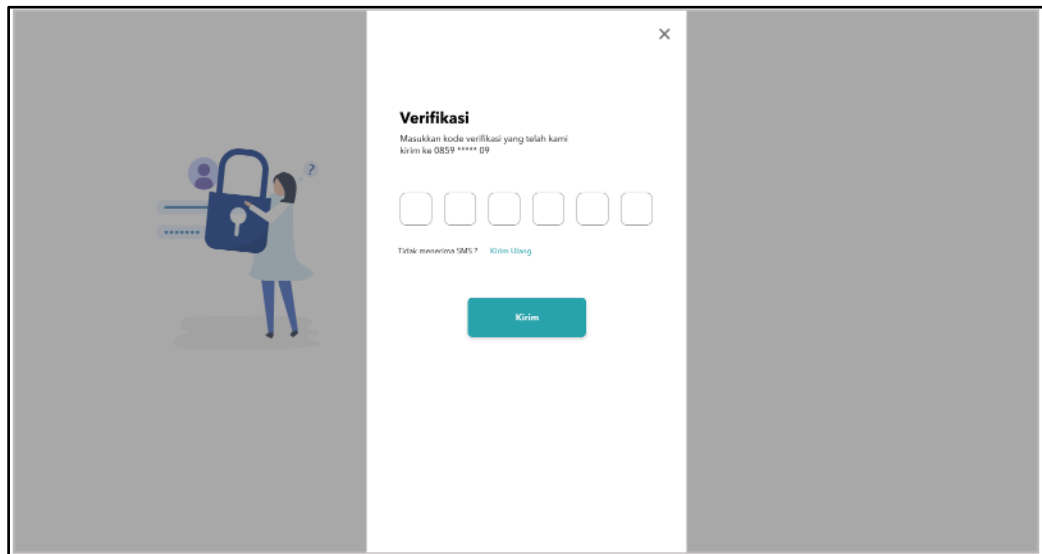
Gambar 3.27 sampai dengan Gambar 3.30 menunjukkan rancangan halaman *forgot password* yang digunakan untuk mereset *password* apabila *user* lupa *password*.



Gambar 3.27 Halaman Mock Up Forgot Password

Gambar 3.27 menggambarkan halaman input nomor telepon pada *Forgot*

Password. Pada halaman ini, *user* diminta untuk memasukkan nomor telepon yang terdaftar sebelumnya untuk melakukan *Forgot Password* lalu menekan *button* Kirim. Jika *user* tidak jadi melakukan proses *forgot password*, maka *user* dapat menekan *button* kembali ke *login*. Setelah *user* melanjutkan proses *forgot password*, *user* diminta untuk memasukkan Kode OTP yang sudah dikirim ke nomor ponsel yang terdaftar sebelumnya.



Gambar 3.28 Halaman Mock Up Forgot Password (Lanjutan)

Gambar 3.28 menggambarkan tampilan halaman verifikasi kode OTP pada *Forgot Password*. Pada halaman ini, *user* diminta untuk memasukkan kode OTP yang dikirimkan ke nomor ponsel *user* yang telah didaftarkan sebelumnya. Lalu menekan *button* Kirim. Bila *user* tidak menerima Kode OTP, maka *user* dapat menunggu waktu tiga menit sampai *user* dapat menekan *link button* Kirim Ulang, maka secara otomatis Kode OTP akan dikirimkan kembali ke nomor ponsel *user* yang terdaftar.

Buat Password Baru

Buat password baru minimal 8 karakter dalam kombinasi huruf dan angka

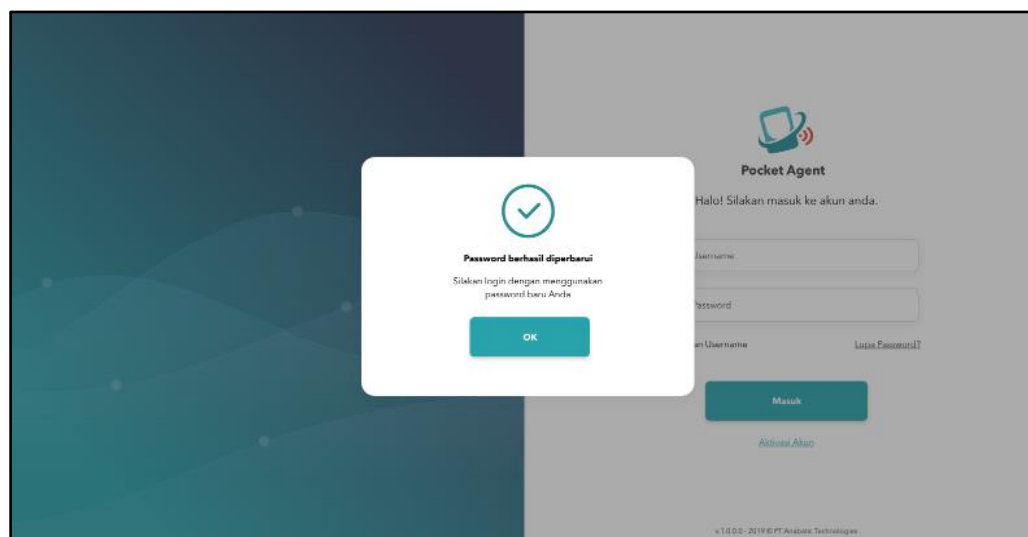
Password Baru *

Ketik Ulang Password Baru *

Kirim

Gambar 3.29 Halaman Mock Up Forgot Password (Lanjutan)

Gambar 3.29 menggambarkan tampilan *form* untuk membuat *password* baru pada *forgot password* dimana pada halaman ini, setelah memasukkan kode OTP valid yang dikirimkan, *user* baru dapat memasukkan *password* baru sesuai yang diinginkan lalu *user* juga wajib untuk memasukkan *password* baru yang sama untuk melakukan konfirmasi *password* baru lalu *user* dapat menekan *button* Kirim dan secara otomatis, *password* berhasil diganti.



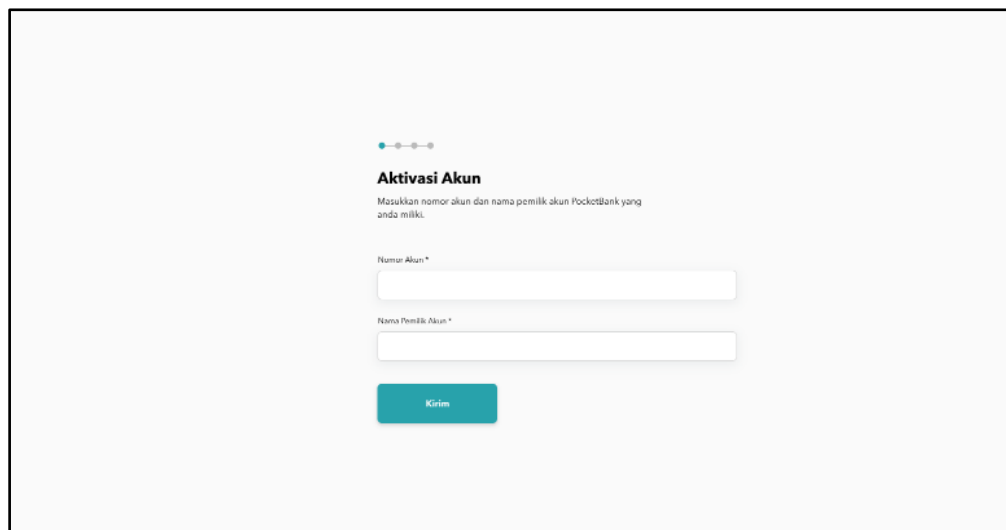
Gambar 3.30 Halaman Mock Up Forgot Password (Lanjutan)

Gambar 3.30 menggambarkan *alert notification* ketika *forgot password*

sukses. Setelah *user* memasukkan *password* baru sebelumnya, maka aplikasi akan mengeluarkan notifikasi kalau *password* yang telah digunakan dan *user* sudah dapat menggunakan *password* yang baru untuk melakukan *login process*.

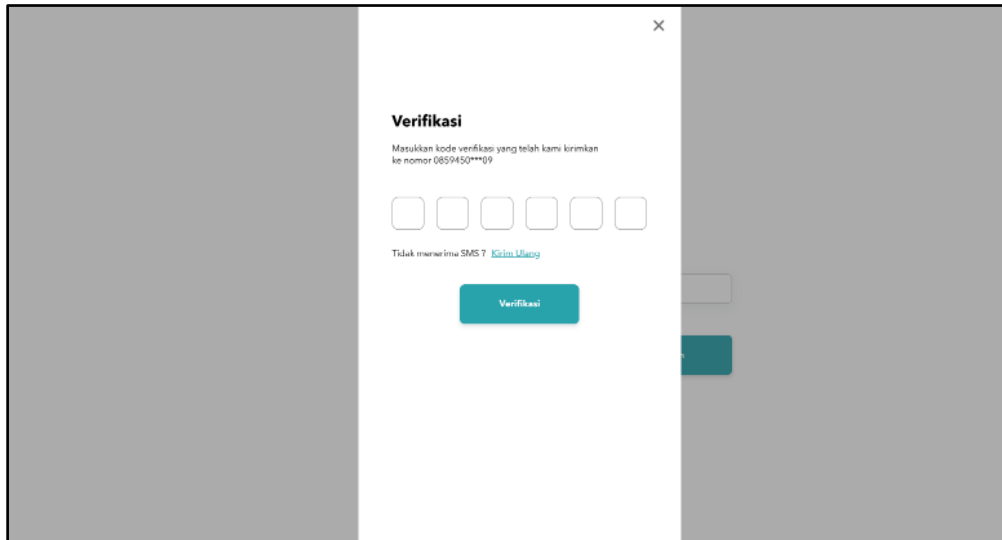
3. Halaman Mock Up Activation Account

Gambar 3.31 sampai dengan Gambar 3.37 menunjukkan halaman *activation account* yang digunakan mengaktivasi akun user untuk menggunakan aplikasi Pocketbank Agent.



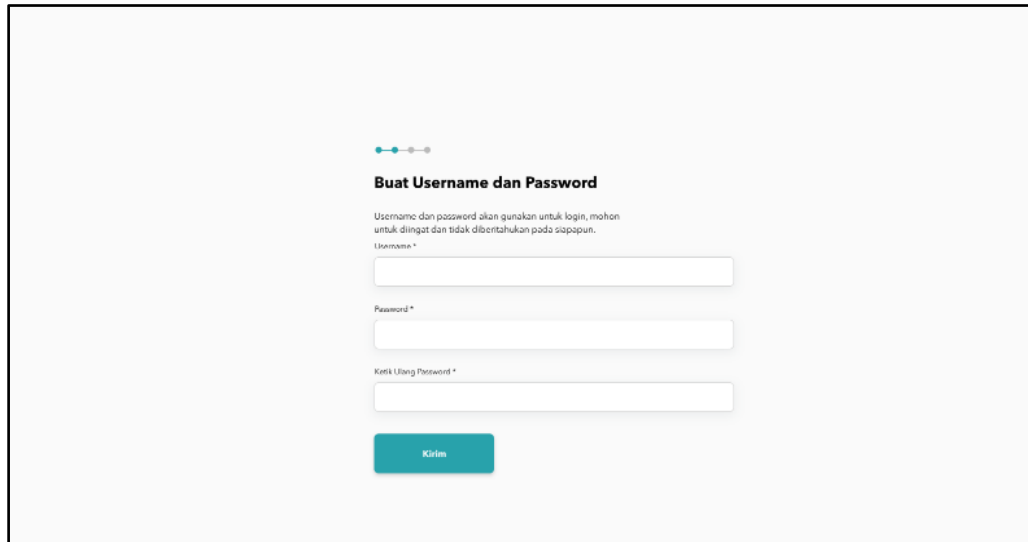
Gambar 3.31 Halaman Mock Up Activation Account

Gambar 3.31 menggambarkan halaman *activation account* untuk mendaftarkan nomor ponsel dan nama lengkap pemilik akun sebagai proses pertama dalam melakukan aktivasi akun. Pada halaman ini, *user* diminta untuk memasukkan nama lengkap dan nomor telepon yang hendak didaftarkan lalu *user* dapat menekan *button* Kirim.



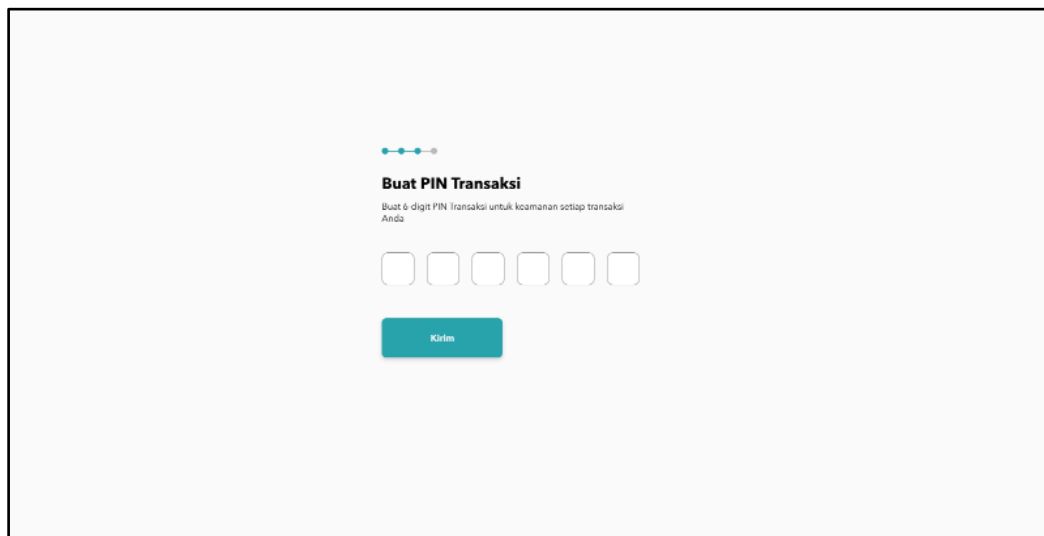
Gambar 3.32 Halaman Mock Up Activation Account (Lanjutan)

Pada Gambar 3.32 menggambarkan *modal* verifikasi kode OTP yang dikirimkan ke nomor ponsel *user* yang didaftarkan sebelumnya. Pada halaman ini, *system* secara otomatis mengirimkan kode OTP ke nomor ponsel yang didaftarkan *user* sebelumnya, lalu *user* diminta untuk memasukkan kembali kode OTP yang telah dikirimkan. Setelah itu *user* dapat menekan *button* Kirim dan secara otomatis *system* melakukan validasi kode OTP yang dikirimkan. Jika *user* tidak mendapatkan kode OTP, maka *user* dapat menunggu jeda pengiriman kode OTP yang sebelumnya untuk meminta kode OTP yang baru dengan menekan *link button* Kirim Ulang, maka secara otomatis *system* akan mengirimkan kode OTP yang baru ke nomor ponsel *user*.



Gambar 3.33 Halaman Mock Up Activation Account (Lanjutan)

Gambar 3.33 menggambarkan halaman untuk membuat *username* dan *password* pada halaman *activation account*. Halaman ini terjadi jika kode OTP yang dikirimkan oleh *user valid*, maka *user* diminta untuk memasukkan *username* dan *password* sesuai keinginan lalu *user* dapat menekan *button Kirim*. Semua data *user* disimpan dalam *database* yang dikirimkan dalam bentuk API.

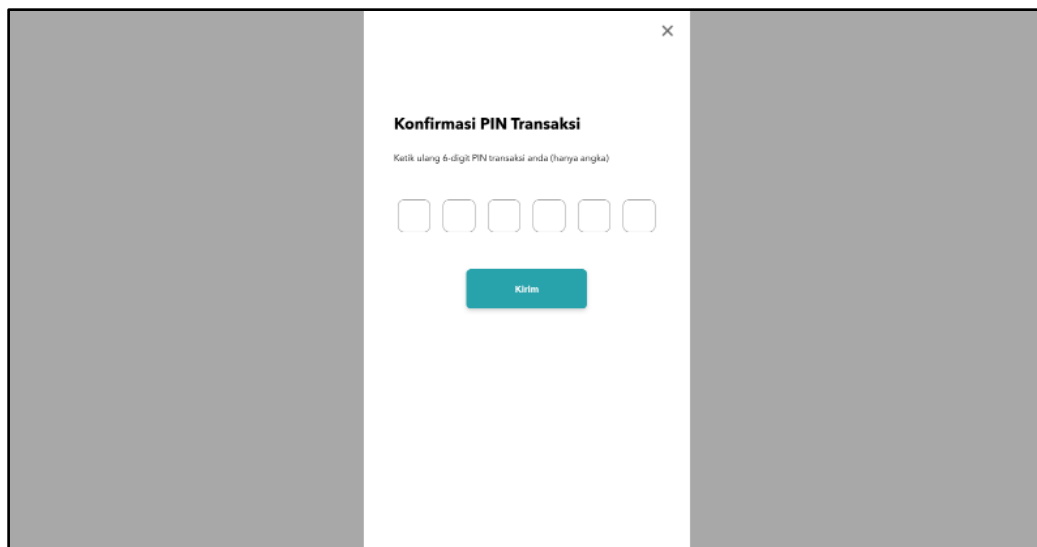


Gambar 3.34 Halaman Mock Up Activation Account (Lanjutan)

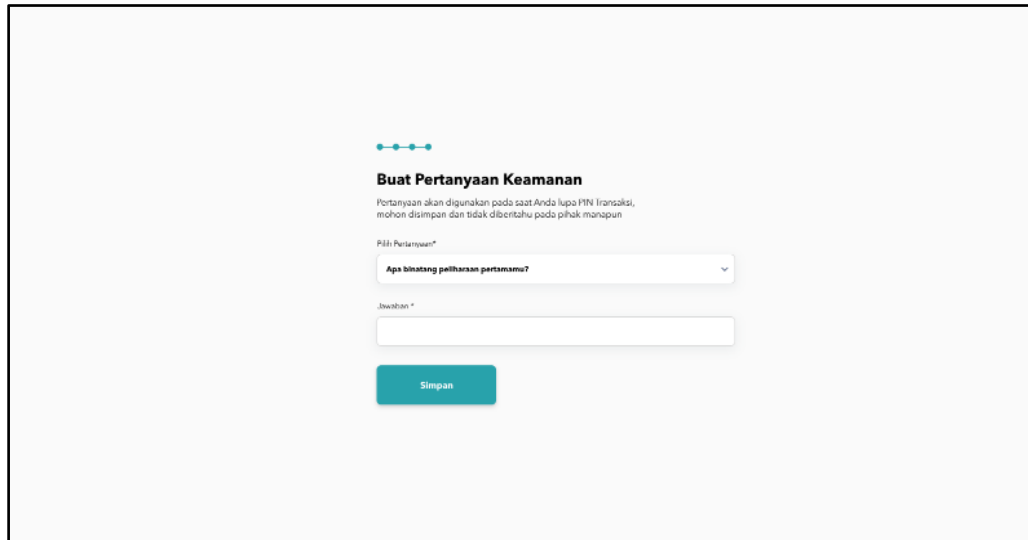
Setelah selesai membuat *username* dan *password* untuk aplikasi, maka *user* diminta untuk memasukkan PIN transaksi dengan mengetik sebanyak 6 digit angka.

PIN transaksi dapat digunakan selama proses semua transaksi yang ada pada Aplikasi Pocketbank Agent. Gambar 3.34 menggambarkan halaman untuk membuat PIN transaksi pada *activation account*.

Gambar 3.35 menggambarkan halaman untuk konfirmasi PIN transaksi yang sudah dimasukkan sebelumnya. Pada halaman ini *user* diminta untuk melakukan konfirmasi PIN transaksi yang dibuat sebelumnya dengan tujuan agar PIN transaksi yang dimaksudkannya benar. Setelah selesai memasukkan PIN transaksi kembali, *user* menekan *button* Kirim. Secara otomatis aplikasi akan melakukan validasi untuk memeriksa kesamaan pengetikan terhadap PIN transaksi yang baru dibuat.

The image shows a mobile app interface for PIN confirmation. It features a white central panel with a close button (X) in the top right corner. The title "Konfirmasi PIN Transaksi" is displayed in bold. Below the title, a subtitle reads "Ketik ulang 6-digit PIN transaksi anda (hanya angka)". There are six empty square input boxes arranged horizontally for the PIN digits. At the bottom of the panel is a teal button labeled "Kirim". The entire white panel is set against a dark gray background.

Gambar 3.35 Halaman Mock Up Activation Account (Lanjutan)



Buat Pertanyaan Keamanan

Pertanyaan akan digunakan pada saat Anda lupa PIN transaksi, mohon disimpan dan tidak diketahui pada pihak manapun

Pilih Pertanyaan*

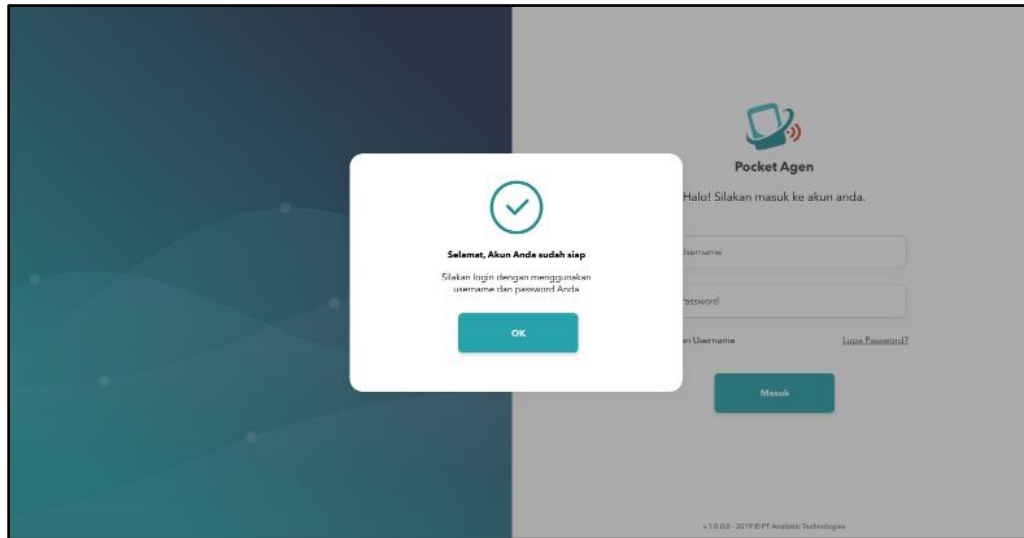
Apa binatang pilihan pertama?

Jawaban *

Simpan

Gambar 3.36 Halaman Mock Up Activation Account (Lanjutan)

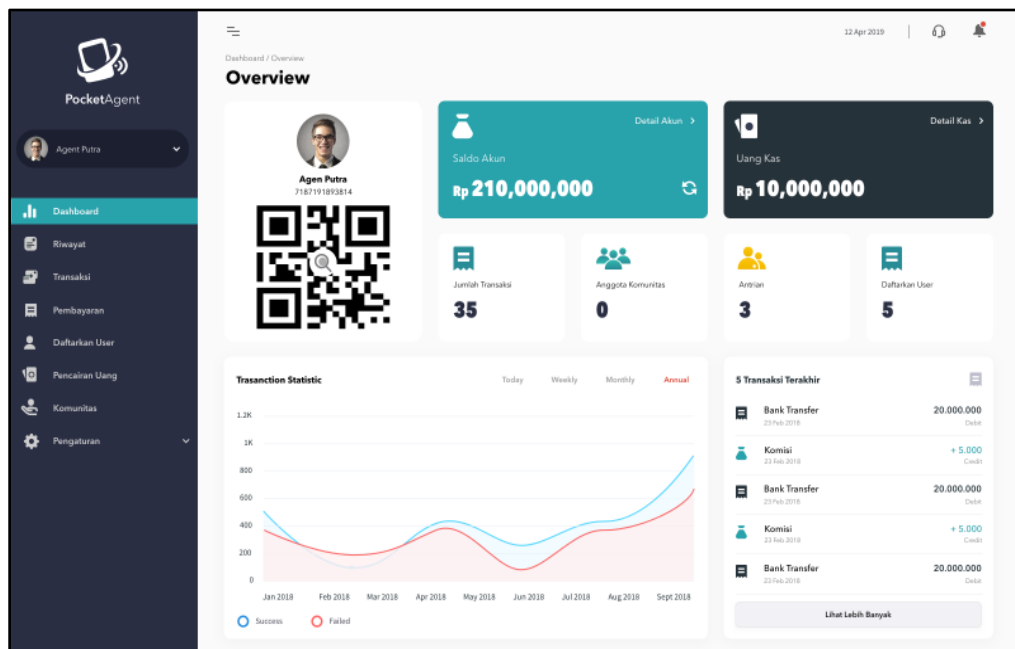
Setelah *user* membuat PIN transaksi, maka pada halaman berikutnya, *user* diminta untuk memilih salah satu *security question* yang berupa pertanyaan sederhana. Setelah memilih *user* diminta juga untuk menjawab *security question* tersebut, lalu *user* dapat menekan *button* Kirim. *Security Question* digunakan untuk meningkatkan keamanan PIN transaksi *user*. Gambar 3.36 menggambarkan halaman untuk memilih dan menjawab *security question* untuk keamanan *account user*. Setelah selesai menjawab *security question*, maka proses pengaktifan akun telah selesai dan secara otomatis aplikasi mengirimkan notifikasi berupa *alert* yang menyatakan bahwa *account* dapat segera digunakan oleh *user*. Gambar 3.37 menggambarkan tampilan ketika *activation account* berhasil dibuat.



Gambar 3.37 Halaman Mock Up Activation Account (Lanjutan)

4. Halaman Mock Up Dashboard

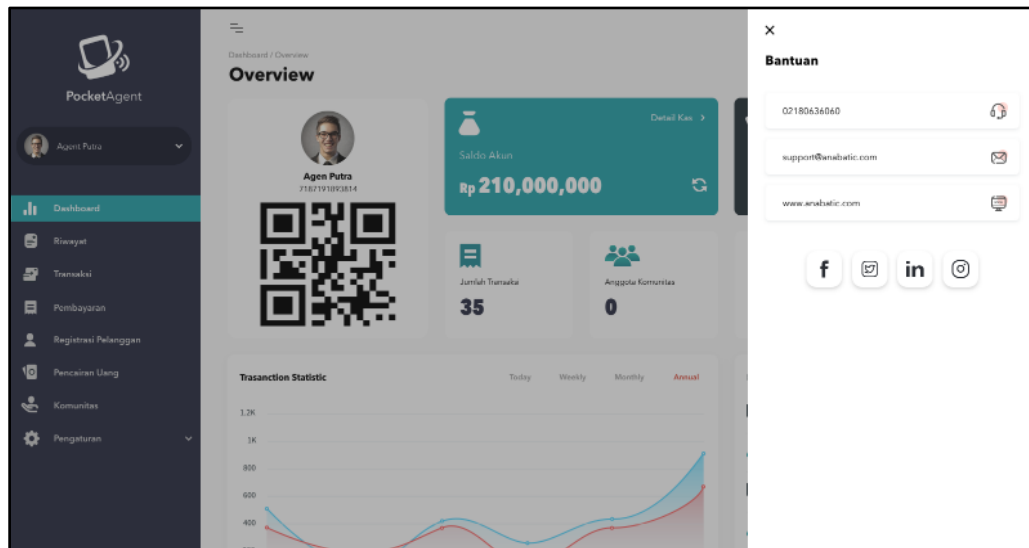
Gambar 3.38 sampai dengan Gambar 3.45 menunjukkan rancangan halaman *dashboard* dimana *user* dapat melihat seluruh *summary* transaksi yang pernah dilakukan pada aplikasi Pocketbank Agent.



Gambar 3.38 Halaman Mock Up Dashboard

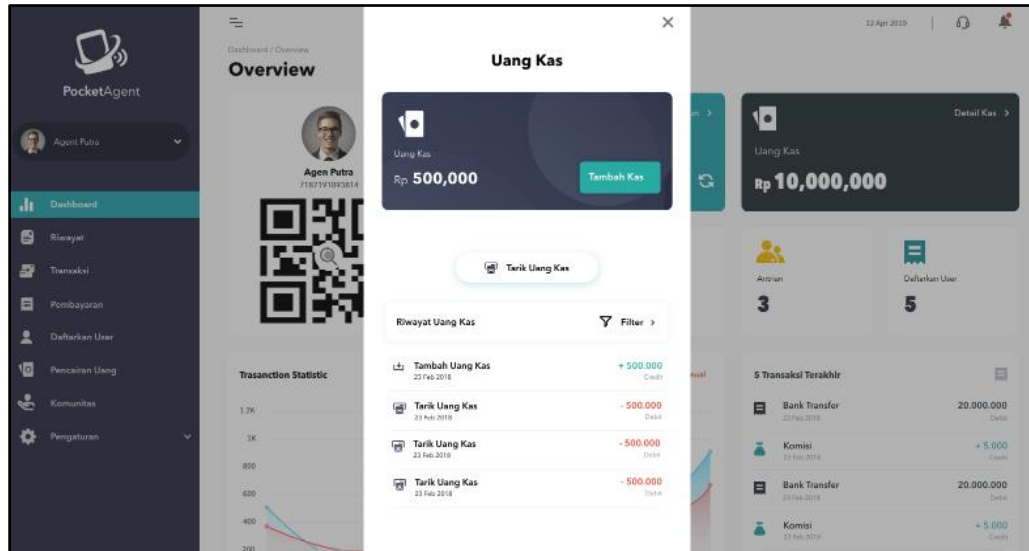
Setelah *login process*, maka *user* langsung masuk ke halaman *dashboard*, Pada halaman *dashboard* terdapat *summary* mengenai *account* dan informasi

mengenai seluruh transaksi yang pernah terjadi. Pada halaman *dashboard*, juga terdapat statistik, dimana *user* dapat mengetahui grafik transaksi yang terjadi baik dalam jangka hari ini,mingguan,bulanan, dan tahunan. Gambar 3.38 menggambarkan tampilan halaman *dashboard* secara keseluruhan.



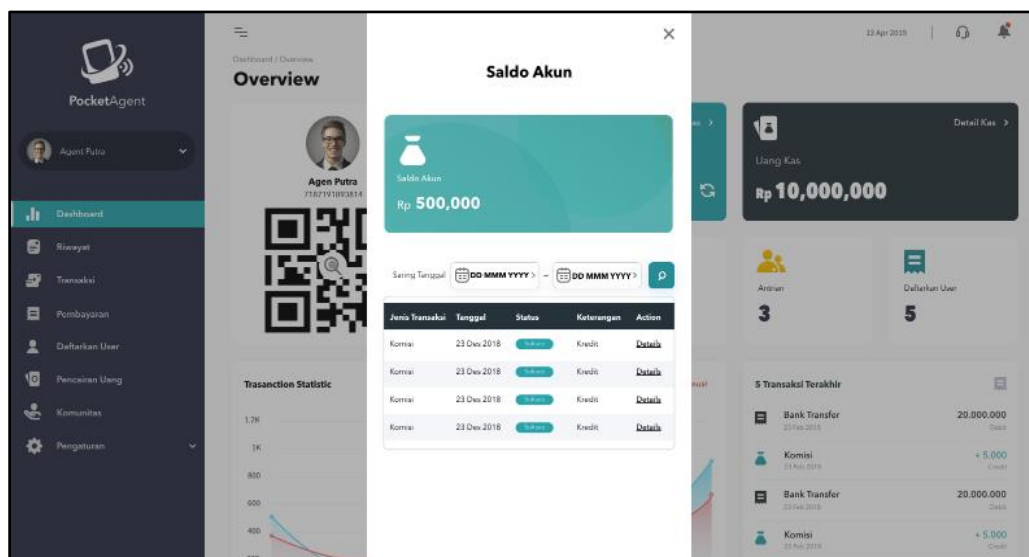
Gambar 3.39 Halaman Mock Up Dashboard (Lanjutan)

Gambar 3.39 menggambarkan tampilan halaman bantuan yang berisi informasi kontak perusahaan. Bagian *sidebar* kanan terdapat menu bantuan, dimana dalam ;menu Bantuan terdapat kontak perusahaan yang terdiri dari nomor telepon perusahaan, *email* perusahaan, *website* perusahaan, dan *social media* beserta *grid* logo *social media* yang dimiliki perusahaan. Fitur ini dapat digunakan oleh *user* bila selama menggunakan aplikasi terdapat kendala dan sebagainya. Fitur *sidebar help* dapat diakses oleh semua fitur yang ada di dalam Aplikasi Pocketbank Agent.



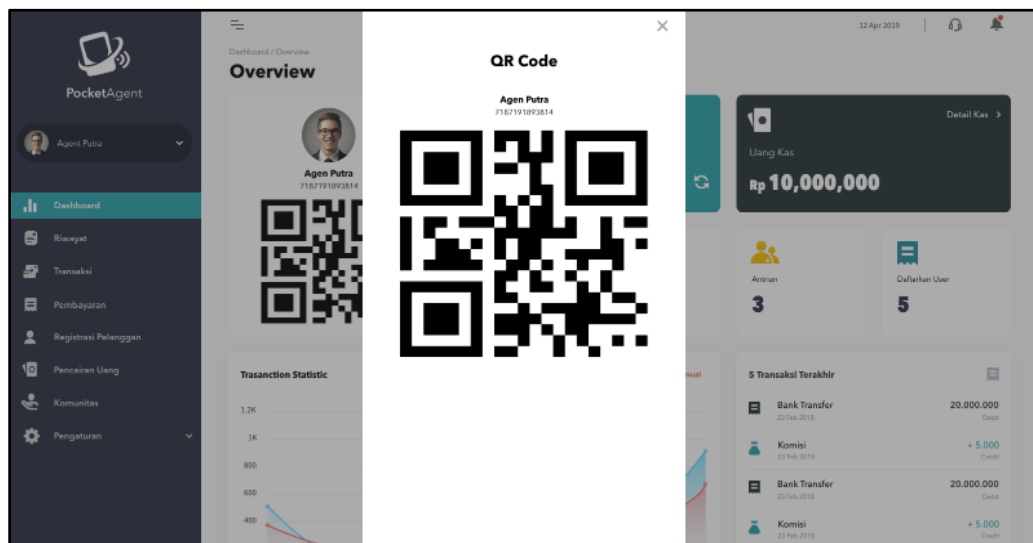
Gambar 3.40 Halaman Mock Up Dashboard (Lanjutan)

Selain Fitur bantuan, *user* juga dapat melihat jumlah uang kas yang dimiliki. Pada *modal detail cash balance*, *user* dapat melihat *detail* transaksi yang pernah terjadi sebelumnya, *user* juga dapat melakukan *filter by date* untuk mempermudah pencarian transaksi. *User* juga dapat melakukan tarik uang kas dan menambah uang kas dengan menekan *button* Tarik Uang Kas atau *button* Tambah Uang Kas. Gambar 3.40 menggambarkan *modal* untuk melihat *detail cash balance*.



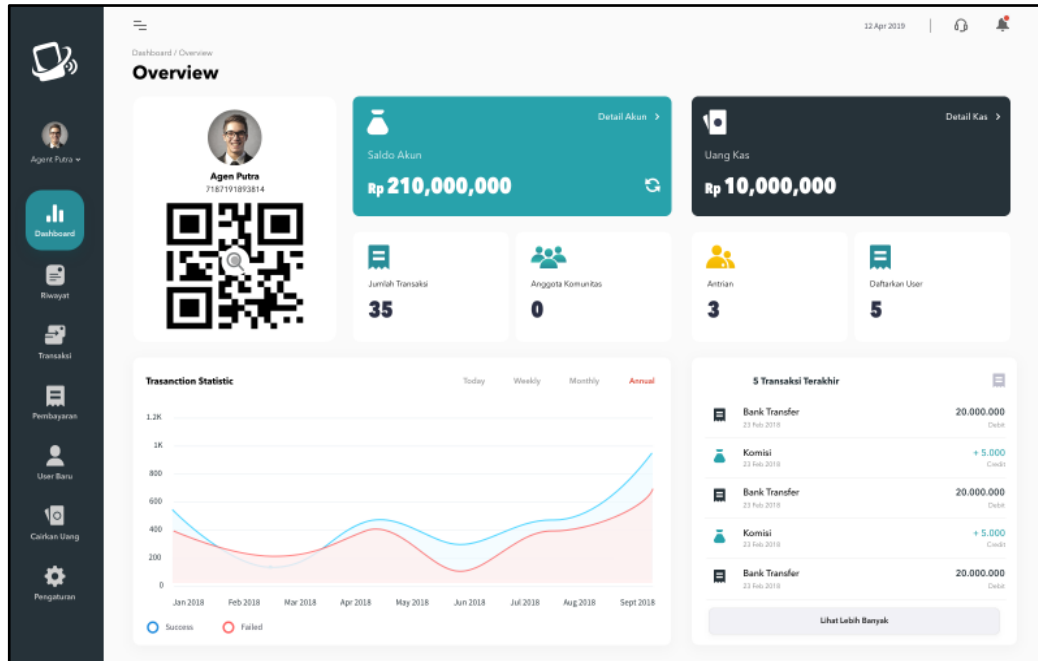
Gambar 3.41 Halaman Mock Up Dashboard (Lanjutan)

Sedangkan Gambar 3.41 menggambarkan *modal* untuk melihat *account balance*. Halaman ini terjadi jika *user* menekan *grid* pada saldo akun. Pada *modal* ini, *user* dapat melihat jumlah saldo akun yang dimiliki, *user* juga dapat melakukan *filter by date* untuk mencari komisi yang masuk dalam akunnya. *User* juga dapat menekan *button detail* pada tabel yang akan diarahkan menuju *Transaction History* yang lebih *detail*.



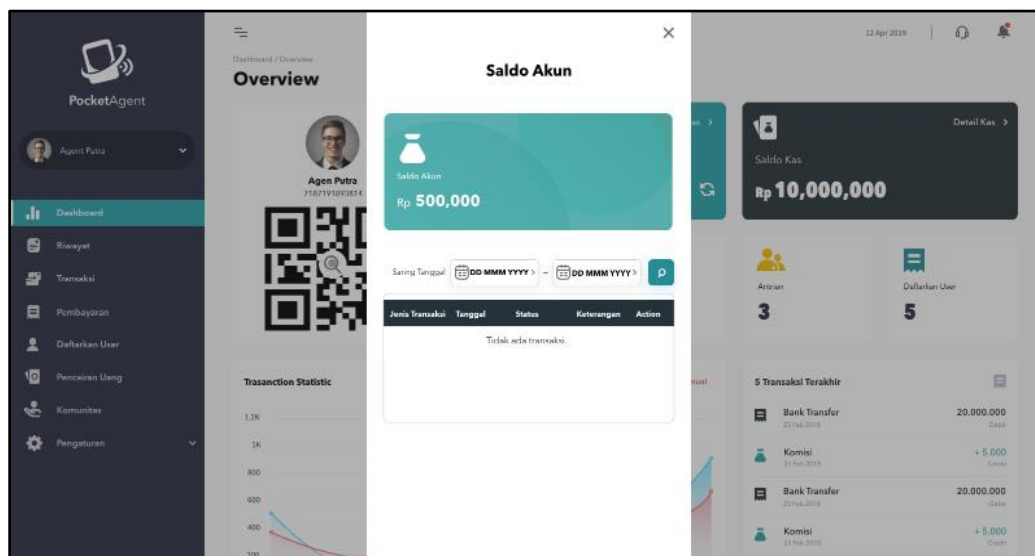
Gambar 3.42 Halaman Mock Up Dashboard (Lanjutan)

Gambar 3.42 menggambarkan *QR Code* sesuai dengan *username* yang dimiliki oleh *user*. Halaman *modal* terjadi jika *user* menekan *grid* pada bagian *username* halaman *dashboard*. Pada *modal* ini terdapat informasi *detail* akun *user* yang berupa *QR Code*. Dalam *modal* terdapat nama lengkap *user* beserta nomor rekening *user*. *QR Code* yang digunakan adalah *QR Code* Standar Indonesia.



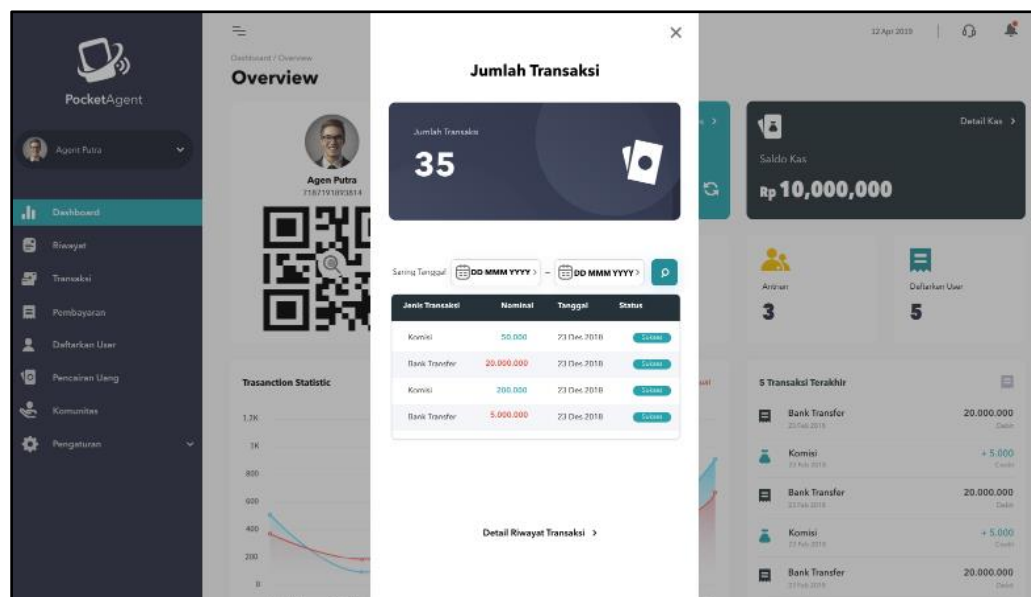
Gambar 3.43 Halaman Mock Up Dashboard (Lanjutan)

Sedangkan Gambar 3.43 menggambarkan tampilan ketika *icon* pada *sidebar* menu dikecilkan. *User* dapat mengecilkan menu *sidebar* Aplikasi Pocketbank Agent dengan menekan *button* menu yang ada diatas *breadcrumb* setiap fitur. Bila *button* menu ditekan, maka tampilan menu *sidebar* hanya akan terdiri dari *icon* setiap fitur yang ditampilkan dan layar pada *dashboard* menjadi agak lebih besar.



Gambar 3.44 Halaman Mock Up Dashboard (Lanjutan)

Gambar 3.44 menggambarkan tampilan ketika belum pernah terjadi transaksi pada *account balance*, dimana pada halaman ini merupakan halaman yang terjadi apabila *user* baru pertama kali *login* dan belum pernah melakukan transaksi apapun dalam Aplikasi Pocketbank Agent. Sehingga tidak ada data pada table transaksi. Namun, pada halaman ini *user* tetap dapat melihat *account balance* apabila ada. Halaman ini terjadi jika *user* menekan saldo akun pada halaman *dashboard*.

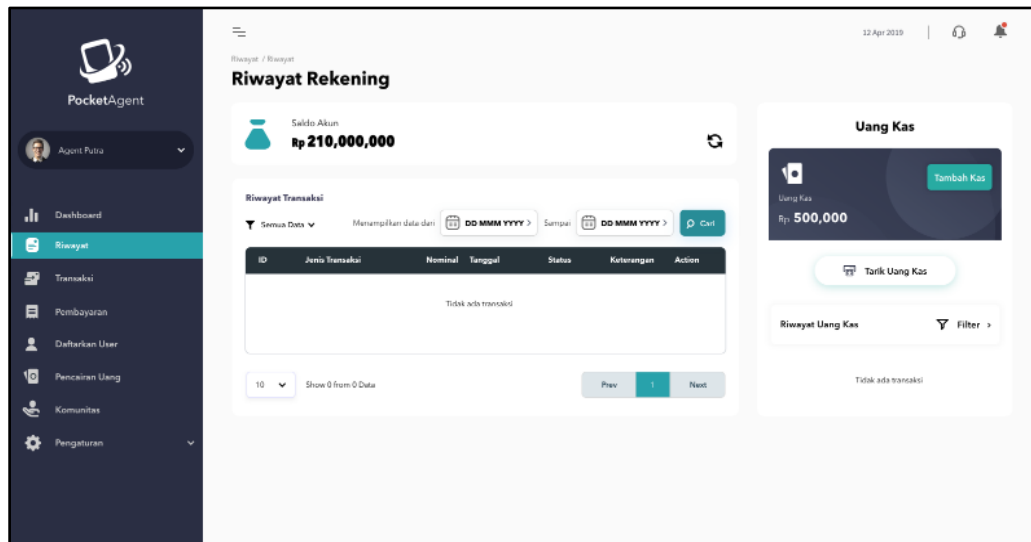


Gambar 3.45 Halaman Mock Up Dashboard (Lanjutan)

Dalam Aplikasi Pocketbank Agent, *user* dapat melihat jumlah transaksi yang pernah dilakukan. Halaman ini terjadi jika *user* menekan jumlah transaksi pada menu *dashboard*. Gambar 3.45 menggambarkan jumlah transaksi yang pernah dilakukan oleh *user*. Pada tabel transaksi dijelaskan semua jenis transaksi yang dilakukan oleh *user* beserta status trnasaksinya. Pada *modal* jumlah transaksi, *user* dapat melakukan *filter by range date* tertentu untuk mempermudah mencari transaksi. Jika *user* menekan *button* Detail Riwayat Transaksi maka *user* akan diarahkan pada bagian fitur History untuk melihat riwayat transaksi lebih *detail*.

5. Halaman Mock Up History

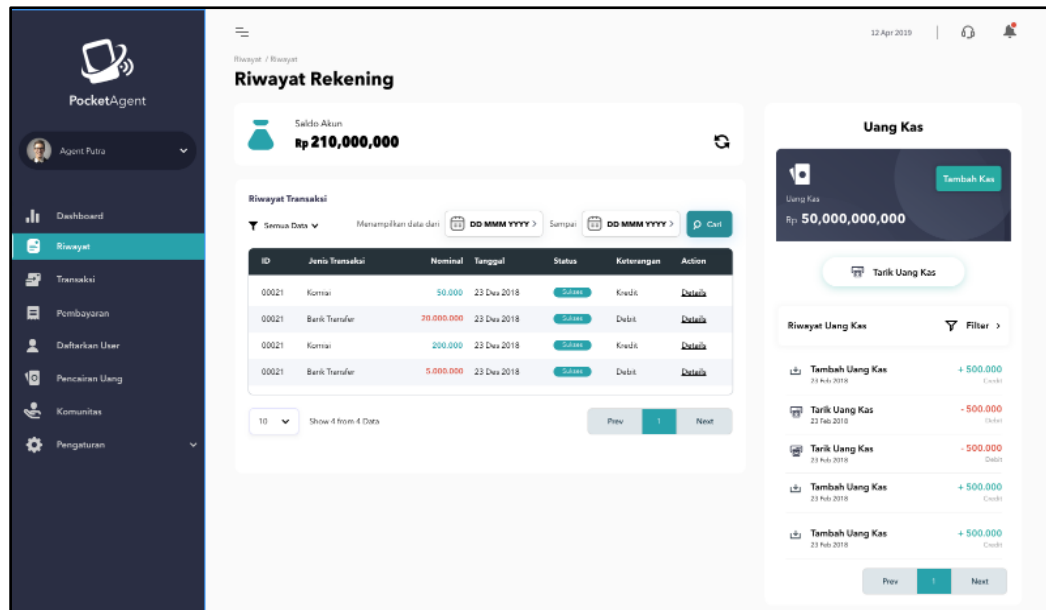
Gambar 3.46 sampai Gambar 3.51 menunjukkan rancangan halaman riwayat Pocketbank Agent yang dapat digunakan oleh *user* untuk melihat riwayat transaksi yang dilakukan melalui aplikasi.



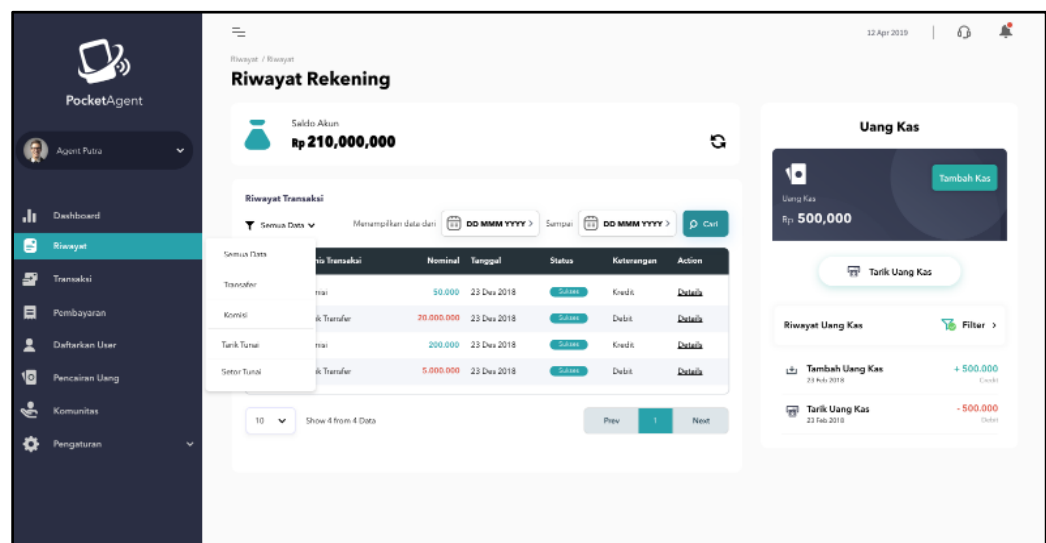
Gambar 3.46 Halaman Mock Up History

User dapat mengakses Halaman History dari menu *sidebar* pada Aplikasi Pocketbank Agent. Pada Halaman History, *user* dapat melihat saldo akun beserta riwayat transaksi secara *detail*. *User* juga dapat melakukan *filter* untuk mempermudah pencarian transaksi pada *range* tanggal tertentu. Pada bagian tabel transaksi, *user* dapat menampilkan *request* jumlah data yang akan ditampilkan pada tabel sesuai dengan pilihan pada *dropdown* pada bagian bawah tabel. Pada bagian *grid* sebelah kanan, terdapat fitur uang kas, dimana *user* dapat menarik dan menambah uang kas serta melihat dan melakukan filter untuk pencarian transaksi. Gambar 3.46 menggambarkan tampilan halaman *history* ketika belum pernah ada transaksi. Sedangkan Gambar 3.47 menggambarkan tampilan halaman *history* ketika *user* sudah melakukan transaksi. Pada halaman ini, *user* dapat menekan *button reload* untuk melakukan reload terhadap fitur History untuk memastikan

bahwa data yang terlihat pada tabel transaksi merupakan data transaksi terbaru yang terjadi pada Aplikasi Pocketbank Agent.



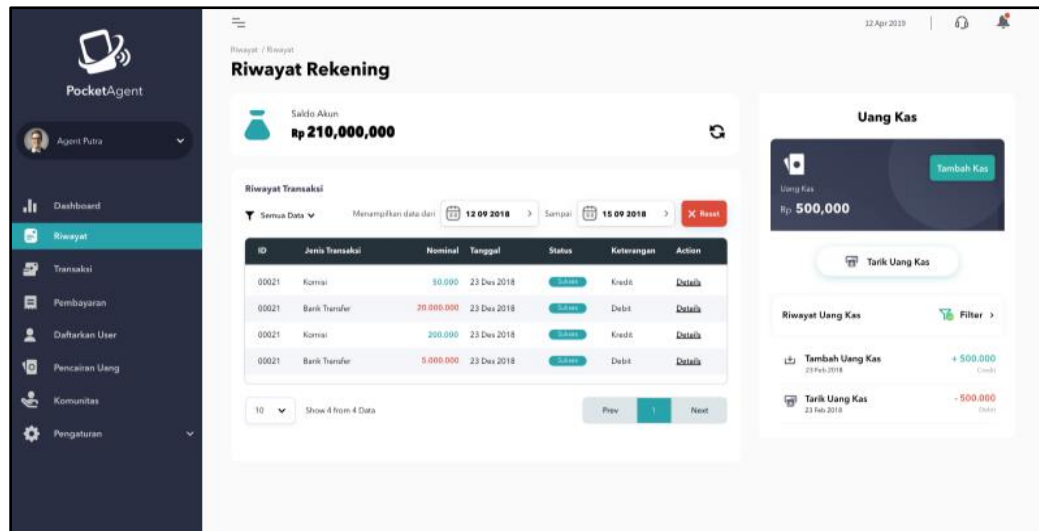
Gambar 3.47 Halaman Mock Up History (Lanjutan)



Gambar 3.48 Halaman Mock Up History (Lanjutan)

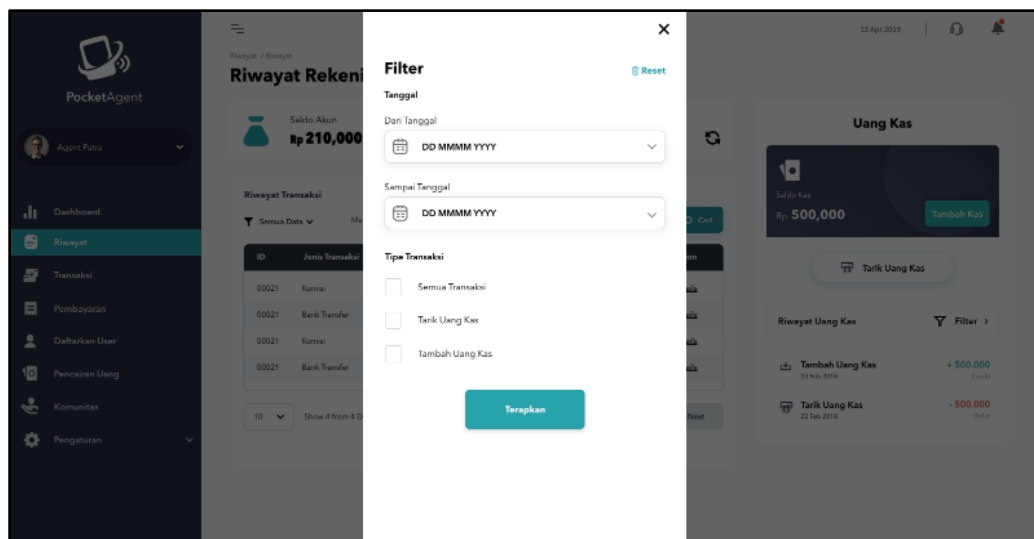
Gambar 3.48 menggambarkan pilihan *filter* pada tabel *history*. Pada Halaman History terdapat *button* berlogo *filter*. Halaman History memiliki *filter* berdasarkan jenis transaksi yang ingin ditampilkan *user* misalnya jika *user* hanya ingin menampilkan jenis transaksi transfer, maka *user* dapat menekan *list* transfer

untuk melakukan *filter* maka secara otomatis, aplikasi akan menampilkan tabel yang sesuai dengan jenis transaksi.



Gambar 3.49 Halaman Mock Up History (Lanjutan)

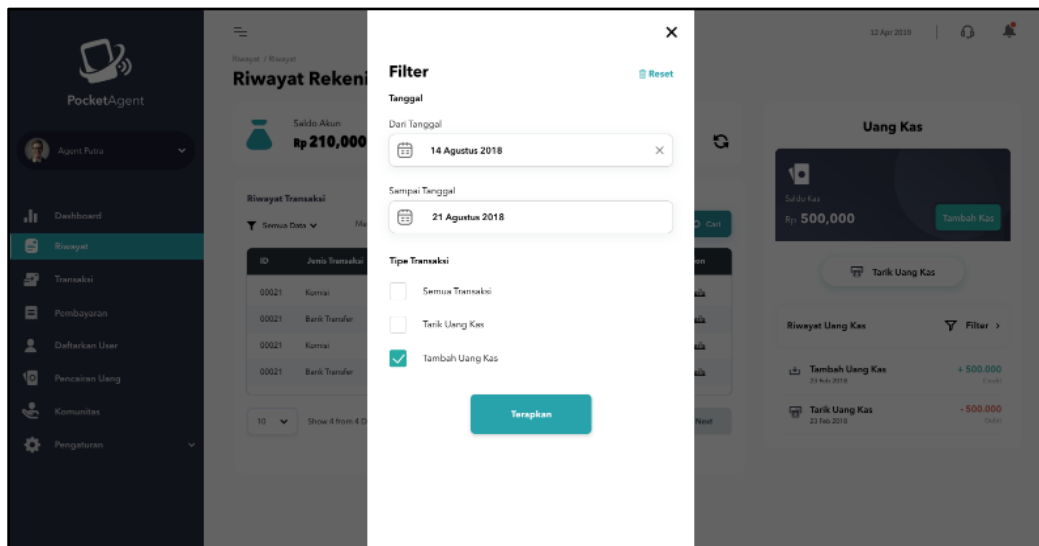
Ketika *user* melakukan *filter* pada jenis transaksi tersebut berdasarkan *range* waktu tertentu pada Tabel Riwayat Transaksi, maka *user* dapat menekan *button Reset* yang berwarna merah disebelah *dropdown* tanggal. Gambar 3.49 menggambarkan *button* untuk melakukan *reset* terhadap *filter*.



Gambar 3.50 Halaman Mock Up History (Lanjutan)

Pada bagian *cash balance*, *user* juga dapat melakukan *filter* berdasarkan

range tanggal dan jenis transaksi yang dilakukan seperti tambah uang kas, tarik uang kas, atau tranfer., Menu Filter dibuat dalam bentuk *open close modal*. User juga dapat melakukan *reset* apabila ingin mengganti *rules filter* pada *cash balance*. Gambar 3.50 menggambarkan *modal* untuk *filter* pada *grid cash balance*.



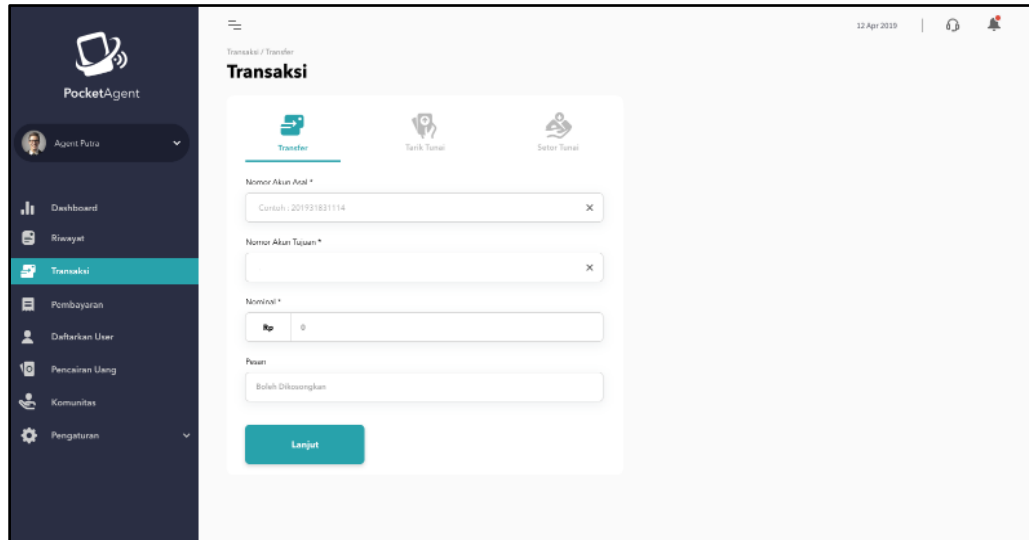
Gambar 3.51 Halaman Mock Up History (Lanjutan)

Pada Gambar 3.51 menggambarkan *tutorial* untuk melakukan *filter* pada bagian *cash balance*. Dimana *user* dapat menekan bagian kolom *dropdown* pada bagian tanggal. *User* dapat memilih *range* tanggal tertentu dan mencentang satu atau lebih dari satu jenis transaksi yang ingin ditampilkan lalu *user* menekan *button* Terapkan untuk melakukan fungsi *filter*. *Button Reset* pada kanan atas *modal* dapat digunakan untuk *reset* kriteria *filter* pada modal.

6. Halaman Mock Up Transaction

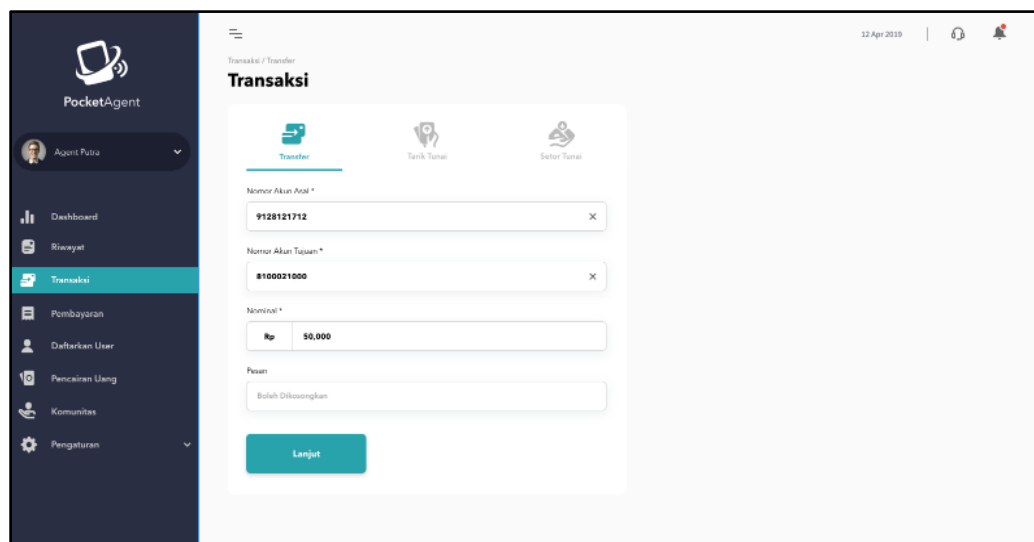
6.1. Halaman Mock Up Transfer

Gambar 3.52 sampai dengan Gambar 3.56 menunjukkan rancangan halaman *Transfer Page* yang merupakan bagian dari halaman *Transaction* yang dapat digunakan *user* untuk melakukan *transfer* antar rekening.



Gambar 3.52 Halaman Mock Up Tranfer

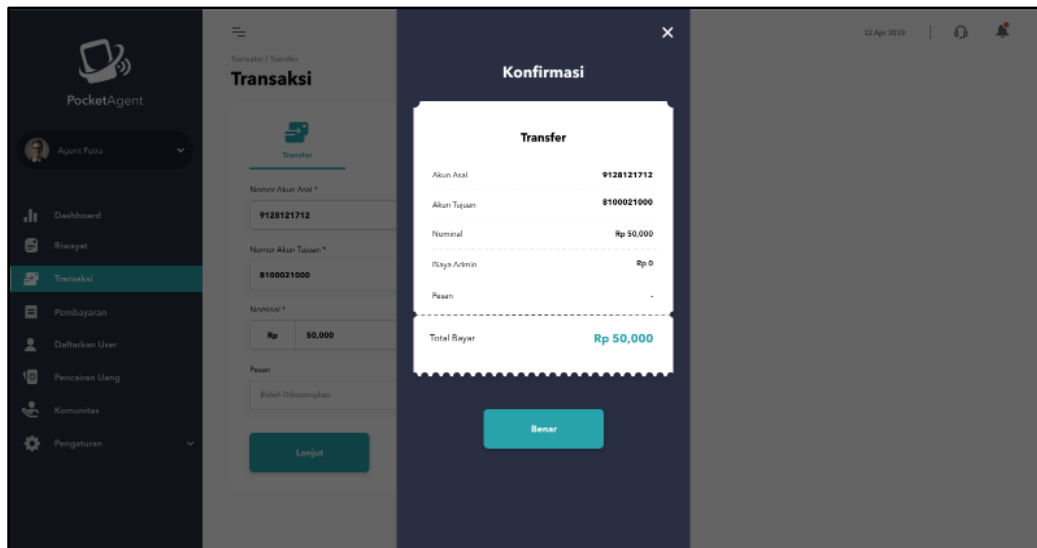
Halaman Transfer terjadi jika *user* memilih Menu Transaksi pada *sidebar menu* Aplikasi Pocketbank Agent. Pada halaman ini terdapat tiga buah *button tab bar* salah satunya *button tab* Halaman Transfer. Pada Halaman Transfer dapat digunakan *user* untuk melakukan transfer kepada orang lain sesuai dengan jumlah nominal tertentu yang diinginkan oleh *user*. Gambar 3.52 menggambarkan tampilan menu *transfer* pada Fitur *Transaction*.



Gambar 3.53 Halaman Mock Up Tranfer (Lanjutan)

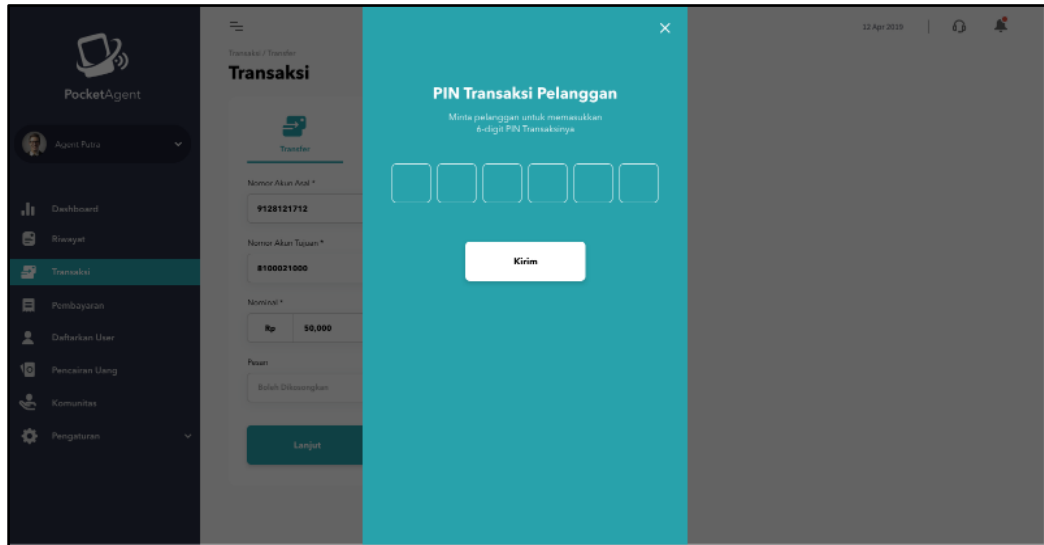
User dapat melakukan transfer dengan mengisi *form* transfer yang terdiri

dari nomor akun asal (nomor akun *user*), nomor akun tujuan, dan nominal yang diinginkan yang merupakan tiga hal yang wajib diisi oleh *user* untuk melakukan proses transfer dan mengisi *textbox* deskripsi secara opsional bila diperlukan. Gambar 3.53 menggambarkan tampilan menu *transfer* ketika *form* sudah diisi.



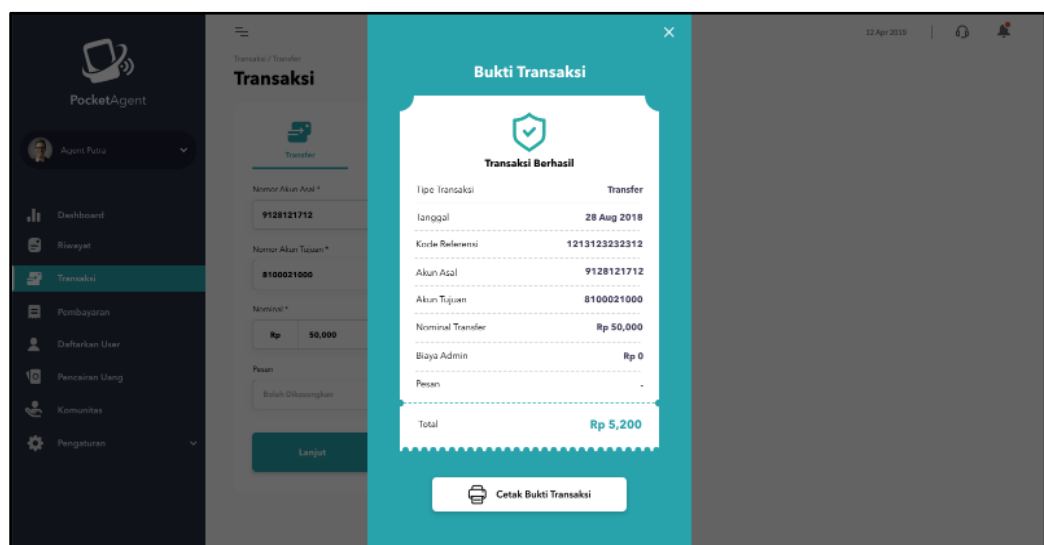
Gambar 3.54 Halaman Mock Up Tranfer (Lanjutan)

Setelah mengisi *form* pada *tab bar* transfer dan menekan *button* Kirim, maka secara otomatis, aplikasi akan mengeluarkan *modal confirmation* yang sesuai dengan *data input user* sebelumnya. Hal ini bertujuan untuk memastikan bahwa data yang dimasukkan oleh *user* sebelum melakukan transaksi benar sehingga mengurangi terjadinya masalah kesalahan dalam proses transfer. Setelah *user* memastikan data yang dimasukkan benar, maka *user* dapat menekan *button* Benar untuk melanjutkan proses transfer. Gambar 3.54 menggambarkan *modal* untuk konfirmasi setelah melakukan pengisian *form transfer*.



Gambar 3.55 Halaman Mock Up Tranfer (Lanjutan)

Setelah menekan *button* Benar pada proses *modal confirmation*, maka *user* diminta untuk memasukkan PIN transaksi yang pernah dibuat ketika mengaktifkan akun. Setelah itu *user* menekan *button* Kirim untuk melanjutkan proses transfer. Bila PIN yang dimasukkan salah, maka secara otomatis aplikasi akan mengeluarkan *alert* yang menyatakan bahwa PIN yang dimasukkan salah. Gambar 3.55 menggambarkan tampilan untuk memasukkan PIN transaksi untuk melakukan transfer.

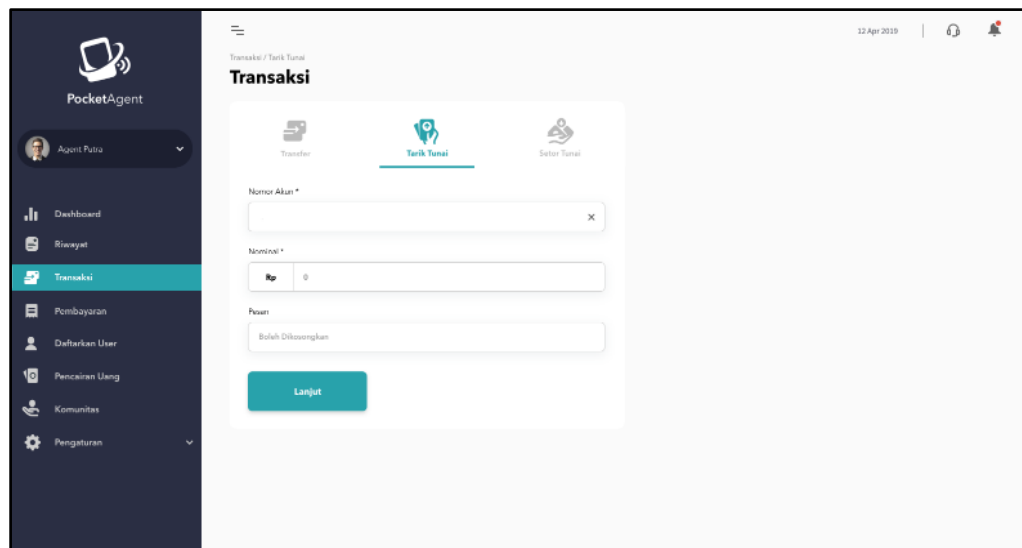


Gambar 3.56 Halaman Mock Up Tranfer (Lanjutan)

Setelah memasukkan PIN transaksi, maka aplikasi akan mengeluarkan *modal* yang menyatakan bahwa transaksi yang dilakukan berhasil. *Modal* tersebut terdapat *button* untuk melakukan *print* dalam bentuk pdf dimana *user* dapat mencetak *invoice* sebagai bukti transfer yang sah. Gambar 3.56 menggambarkan *invoice* untuk menyatakan proses *transfer* telah berhasil dilakukan.

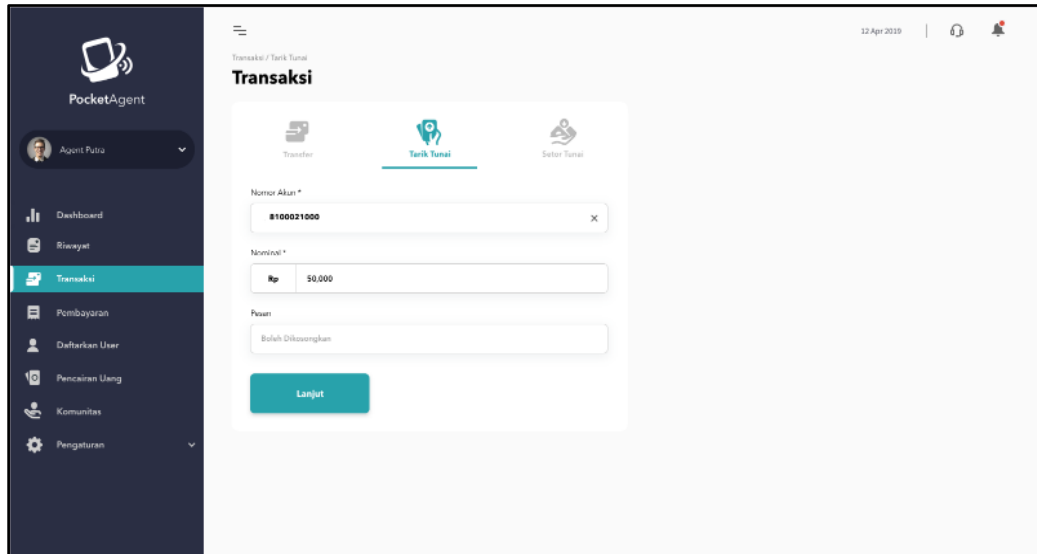
6.2. Halaman Mock Up Cash Out

Gambar 3.57 sampai dengan Gambar 3.61 menunjukkan rancangan halaman *Cash Out Page* yang merupakan bagian dari halaman *Transaction* yang dapat digunakan *user* untuk melakukan tarik tunai.

The image shows a mobile application interface for 'PocketAgent'. On the left is a dark blue sidebar menu with icons and labels for 'Dashboard', 'Riwayat', 'Transaksi' (highlighted), 'Pembayaran', 'Daftar User', 'Pencarian Uang', 'Komunitas', and 'Pengaturan'. The main area is titled 'Transaksi / Tarik Tunai' and 'Transaksi'. It features three tabs: 'Transfer', 'Tarik Tunai' (active), and 'Setor Tunai'. Below the tabs is a form with fields for 'Nomor Akun *' (with a clear button), 'Nominal *' (with a currency selector set to 'Rp' and a value of '0'), and 'Pesan' (with a placeholder 'Silahkan Diisi'). A green 'Lanjut' button is at the bottom of the form. The top right corner shows the date '12 Apr 2019' and notification icons.

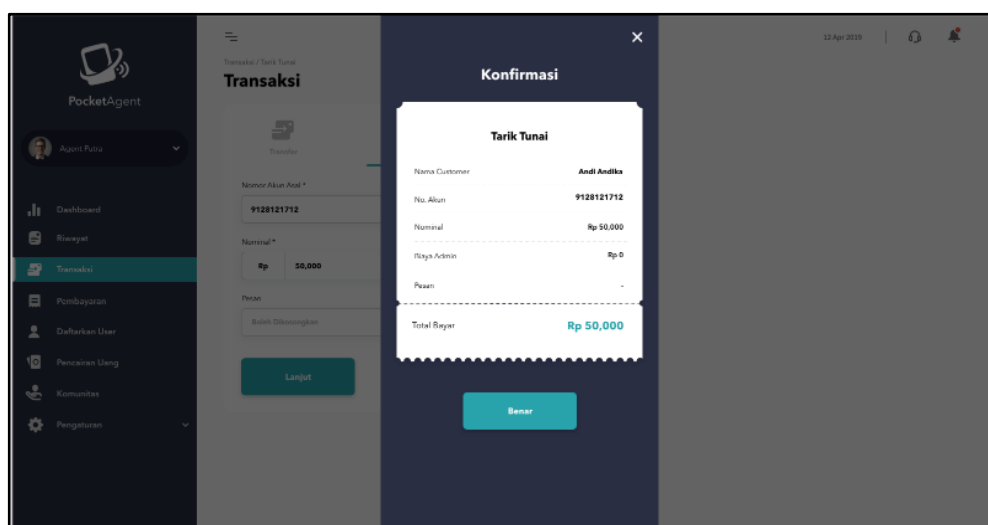
Gambar 3.57 Halaman Mock Up Cash Out

Selain *tab bar* transfer, *user* juga dapat melakukan tarik tunai atau *cash out* pada menu *Transaksi*. Gambar 3.57 menggambarkan tampilan halaman *cash out* pada Fitur *transaction*. *Form cash out* terdiri dari *textbox* Nomor akun, jumlah nominal yang ingin diambil, dan deskripsi.



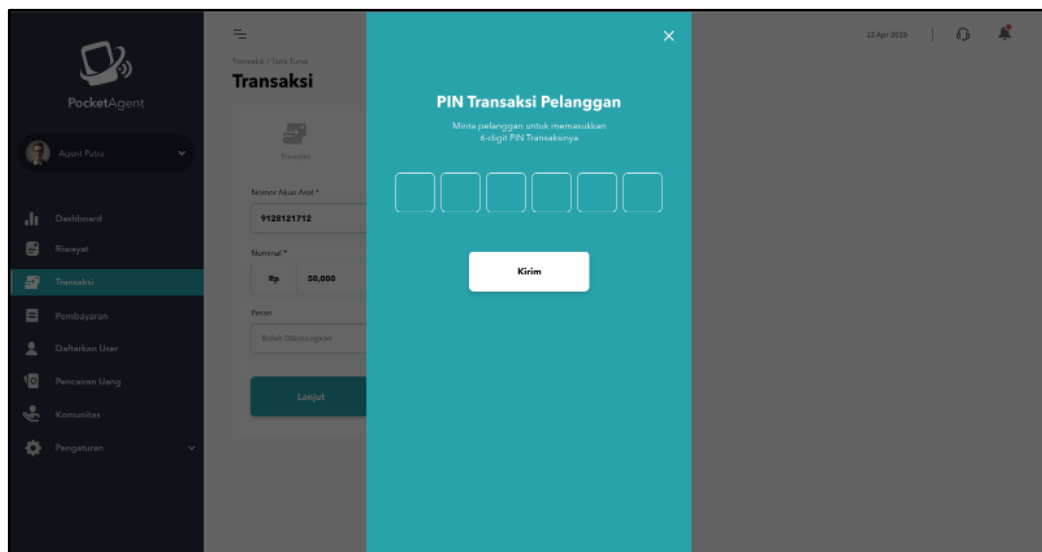
Gambar 3.58 Halaman Mock Up Cash Out (Lanjutan)

Gambar 3.58 menggambarkan *tutorial* mengisi halaman *cash out* ketika *form* tarik tunai sudah diisi. Sama seperti halaman *transfer*, *user* dapat mengisi *form* *cash out* jika ingin melakukan proses tarik tunai, dimana *user* wajib mengisi nomor akun yang akan ditarik tunai dan jumlah nominal yang diinginkan . *User* juga bisa menambahkan deskripsi bila diperlukan (opsional). Jika *user* salah masukkan nomor akun, maka *user* dapat menekan *icon close* yang berbentuk tanda x disebelah kanan *textbox* nomor akun.



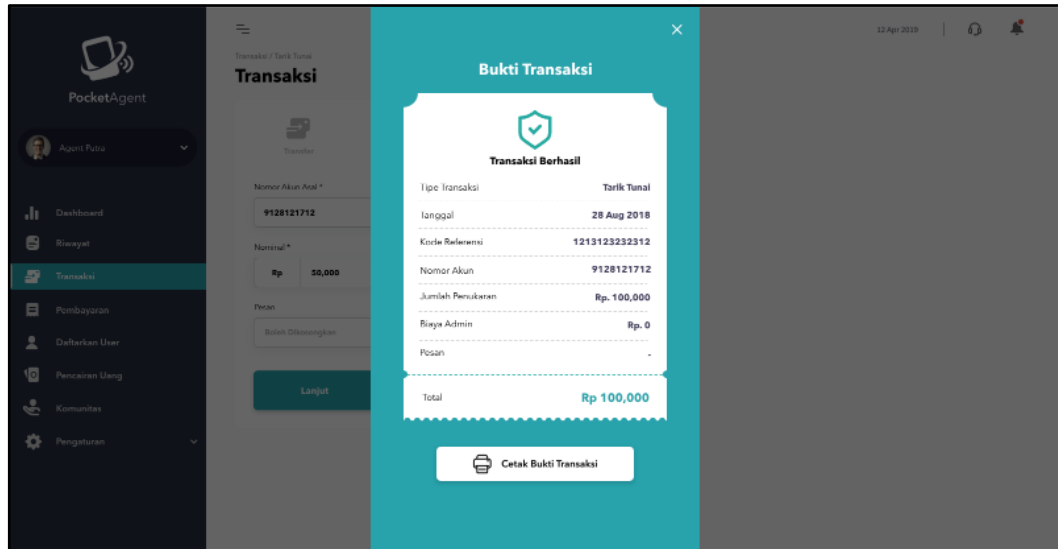
Gambar 3.59 Halaman Mock Up Cash Out (Lanjutan)

Setelah mengisi *form* pada transfer maka aplikasi memberikan respon dengan mengeluarkan *modal confirmation* yang sesuai dengan data *input* user untuk memastikan data yang dimasukkan oleh *user* dengan benar. Setelah *user* memeriksa data yang dimasukkan sebelumnya dengan benar, *user* dapat menekan *button* Benar untuk melanjutkan transaksi tarik tunai. Gambar 3.59 menggambarkan *modal* konfirmasi untuk melakukan tarik tunai.



Gambar 3.60 Halaman Mock Up Cash Out (Lanjutan)

Setelah menekan *button* Benar, maka untuk melanjutkan transaksi, *user* diminta untuk memasukkan 6 digit PIN transaksi yang pernah dibuat ketika melakukan *activation account* lalu menekan *button* Kirim untuk melanjutkan proses transaksi tarik tunai. Jika PIN transaksi yang dimasukkan salah, maka aplikasi akan mengeluarkan *alert notifikasi* yang menyatakan PIN yang dimasukkan salah. Gambar 3.60 menggambarkan tampilan untuk memasukkan PIN transaksi untuk melakukan tarik tunai.



Gambar 3.61 Halaman Mock Up Cash Out (Lanjutan)

Setelah selesai memasukkan PIN transaksi dengan benar, maka proses tarik tunai sudah berhasil dilakukan, maka aplikasi akan mengeluarkan *modal* bukti transaksi tarik tunai beserta dengan waktu terjadinya tarik tunai. Pada *modal* tersebut, *user* dapat mencetak bukti transaksi dengan menekan *button* Cetak Bukti Transaksi sebagai alat bukti transaksi pada aplikasi yang sah. Gambar 3.61 menggambarkan ketika *cash out* berhasil dilakukan.

6.3. Halaman Mock Up Cash In

Gambar 3.62 sampai dengan Gambar 3.66 menunjukkan rancangan halaman *Cash In Page* yang merupakan bagian dari halaman *Transaction* yang dapat digunakan *user* untuk melakukan setor tunai.

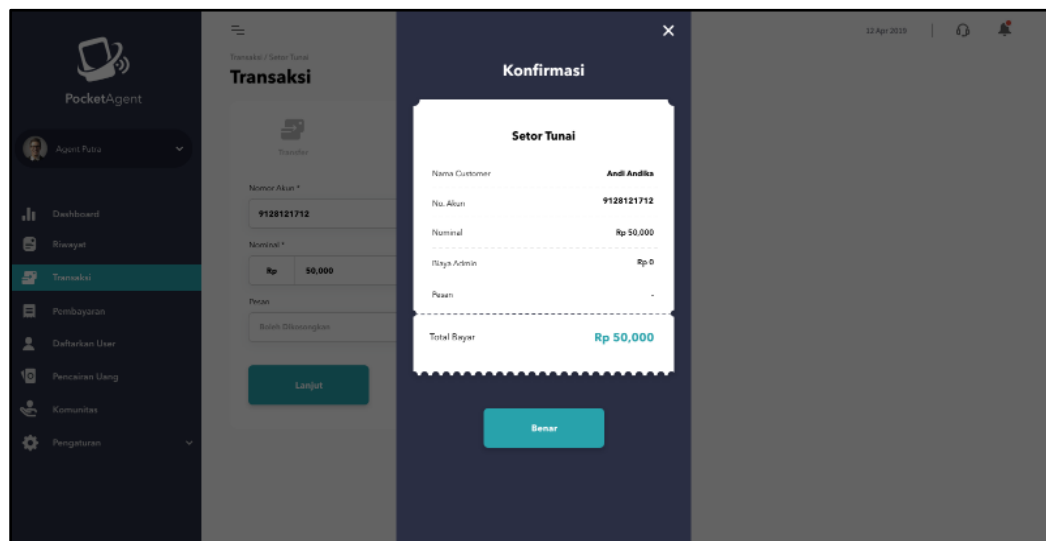
Gambar 3.62 Halaman Mock Up Cash In

Pada menu Transaksi juga terdapat *tab bar cash in*, dimana pada *tab bar cash in*, *user* dapat melakukan setor tunai. Sama seperti *tab bar Transfer* dan *Cash Out*, *tab bar cash in* juga berisi *form* yang tidak jauh berbeda dengan *tab bar* yang lain. *Form Cash In* terdiri dari *textbox* nomor akun, nominal, dan deskripsi yang harus dilengkapi oleh *user* untuk melakukan proses transaksi setor tunai. Gambar 3.62 menggambarkan halaman *cash in* pada Fitur *transaction*.

Gambar 3.63 Halaman Mock Up Cash In (Lanjutan)

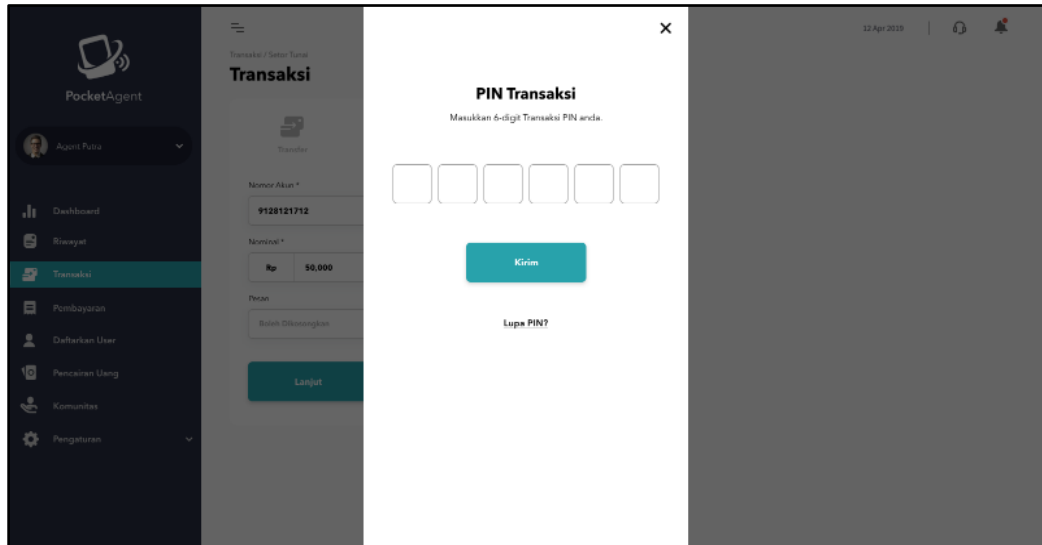
Dalam *tab bar cash in*, *user* wajib mengisi *textbox* nomor akun dan nominal

yang diinginkan dan *user* juga dapat menambahkan deskripsi secara opsional untuk melakukan proses *cash in*., lalu *user* dapat menekan *button* Kirim untuk melanjutkan proses transaksi setor tunai. Bila *user* salah menekan nomor akun, maka *user* dapat menekan *icon* silang pada bagian sebelah kanan *textbox* nomor akun untuk menghapus sekaligus. Gambar 3.63 menggambarkan halaman *cash in* ketika *form cash in* telah diisi.



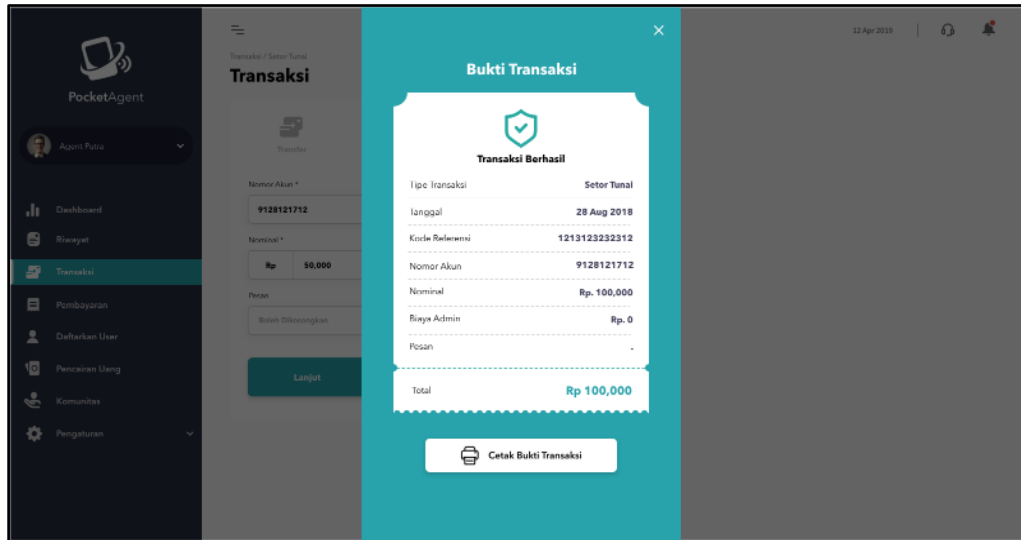
Gambar 3.64 Halaman Mock Up Cash In (Lanjutan)

Setelah menekan *button* Kirim, maka *user* diminta untuk memeriksa kembali data yang telah diinput sebelumnya dan memastikan bahwa nama *customer* pada *modal confirm* benar. Setelah memeriksa, maka *user* dapat menekan *button* Benar untuk melanjutkan proses transaksi. Gambar 3.64 menggambarkan *modal* konfirmasi untuk melakukan setor tunai.



Gambar 3.65 Halaman Mock Up Cash In (Lanjutan)

Setelah menekan *button* Benar, maka *user* diminta untuk memasukkan 6 digit PIN transaksi untuk melanjutkan proses transaksi setor tunai. Bila *user* lupa PIN transaksi, maka *user* dapat menekan *button* Lupa PIN, maka *user* akan diarahkan pada halaman Forgot PIN. Bila *user* hendak membatalkan transaksi, maka *user* dapat menekan *icon close* pada sudut kanan *modal* PIN Transaksi. PIN yang dimasukkan *user* akan diperiksa secara otomatis oleh aplikasi. Apabila PIN yang dimasukkan salah, maka aplikasi akan mengeluarkan *alert notification* yang menyatakan PIN Transaksi yang dimasukkan salah. Gambar 3.65 menggambarkan *modal* untuk memasukkan PIN transaksi untuk melakukan setor tunai.

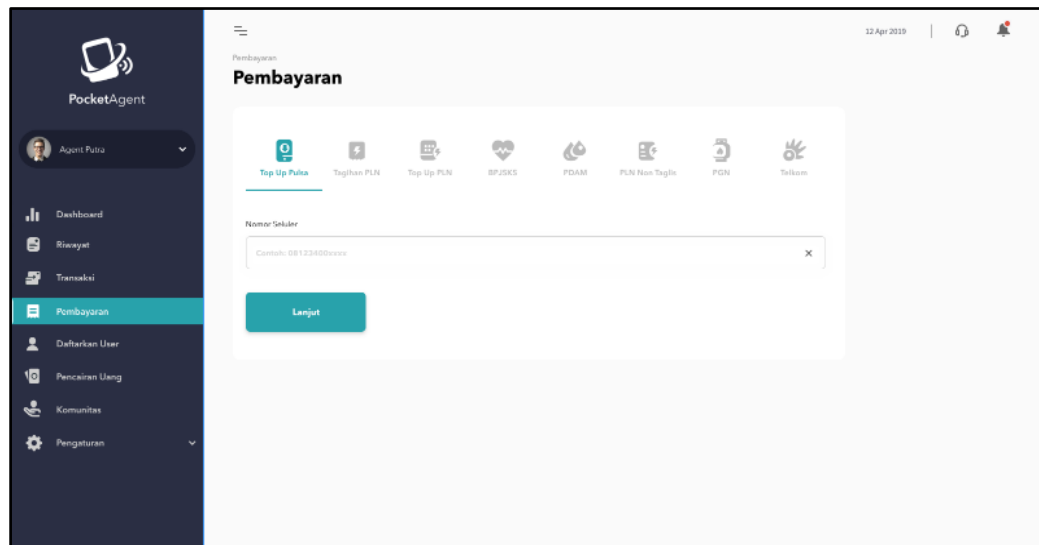


Gambar 3.66 Halaman Mock Up Cash In (Lanjutan)

Setelah selesai memasukkan PIN transaksi, maka aplikasi akan mengeluarkan *modal* selanjutnya yang menyatakan transaksi setor tunai berhasil dilakukan beserta dengan data yang dimasukkan *user* sebelumnya dan waktu transaksi. *User* dapat mencetak bukti transaksi dalam bentuk pdf sebagai bukti sah transaksi pada Aplikasi Pocketbank Agent. Gambar 3.66 menggambarkan ketika proses setor tunai berhasil dilakukan.

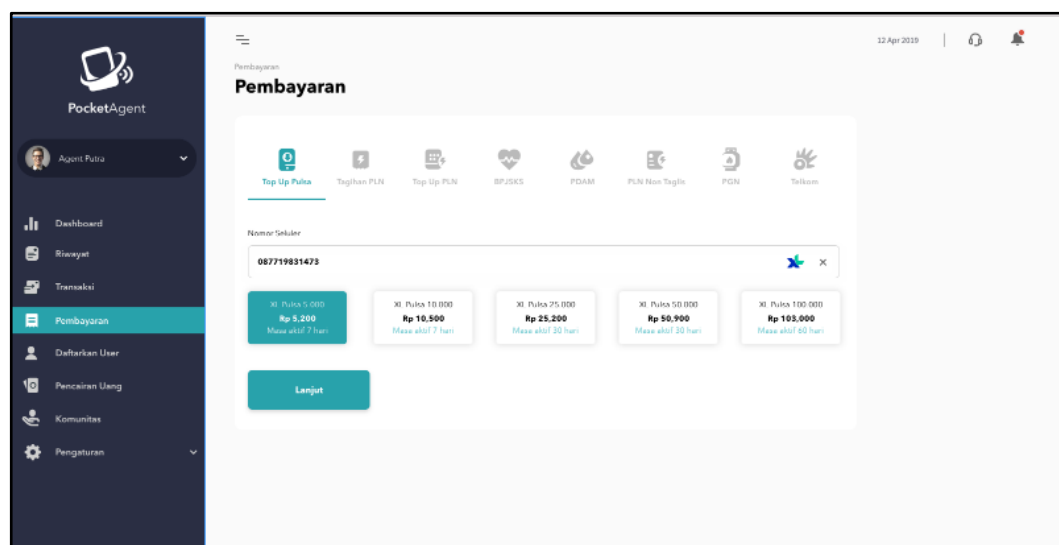
7. Halaman Mock Up Payment (PostPaid Number)

Gambar 3.67 sampai dengan Gambar 3.71 menunjukkan rancangan halaman *Payment Page* yang merupakan bagian dari halaman *Transaction* yang dapat digunakan *user* untuk melakukan membeli *m-commerce* khususnya pada bagian *top up* pulsa.



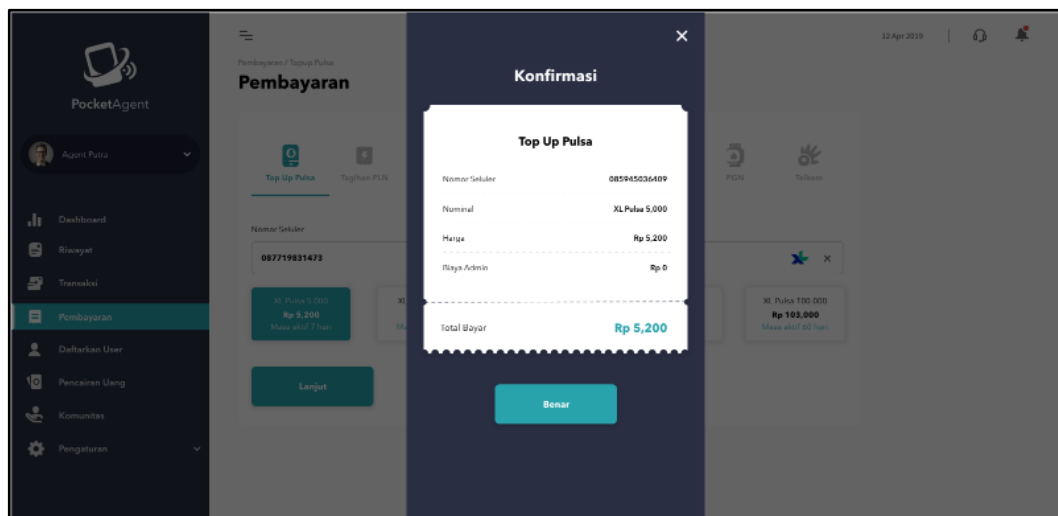
Gambar 3.67 Halaman Mock Up Payment

Pada Aplikasi Pocketbank Agent terdapat fitur untuk membeli produk *m-commerce* seperti *top up*, pembayaran tagihan PLN, *top up* PLN, pembayaran BPJS, pembayaran PDAM, dan lainnya. Untuk melakukan pembelian *product m-commerce*, *user* dapat menekan *tab bar* sesuai dengan jenis *product* yang ingin dibeli dan mengisi *form* sesuai dengan masing masing *product m-commerce*. Gambar 3.67 menggambarkan tampilan untuk menu *payment m-commerce* pada *topup* pulsa.



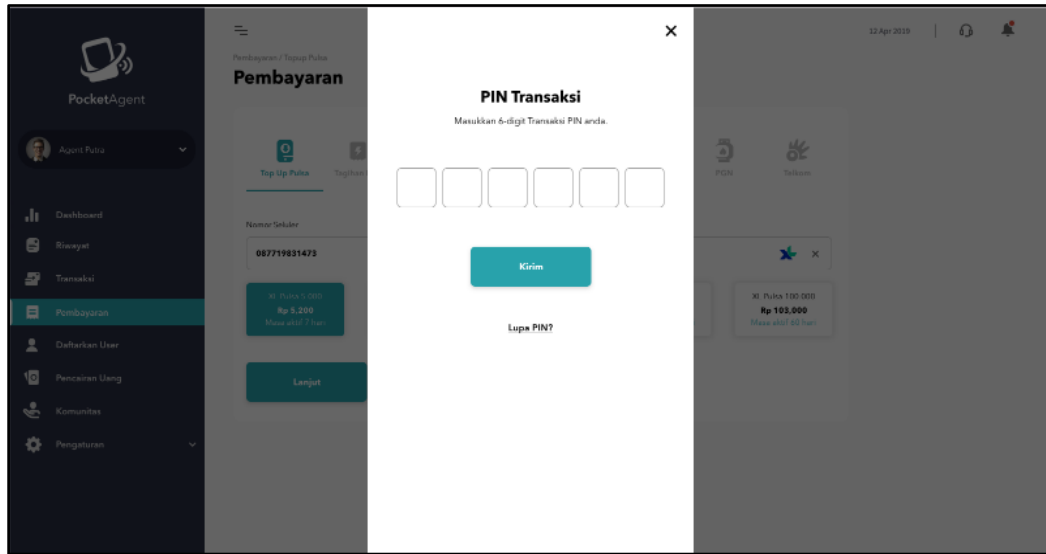
Gambar 3.68 Halaman Mock Up Payment (Lanjutan)

Pada Gambar 3.68 menggambarkan ketika *form m-commerce* pada *topup* pulsa sudah diisi. Dimana untuk melakukan *top up* pulsa, *user* harus memasukkan nomor ponsel yang hendak di *top up*. Aplikasi akan secara otomatis menampilkan logo operator berdasarkan nomor ponsel yang diinput oleh *user* dan mengeluarkan menu *grid top up* pulsa beserta dengan harganya yang dapat langsung dipilih *user* sesuai dengan keinginan.



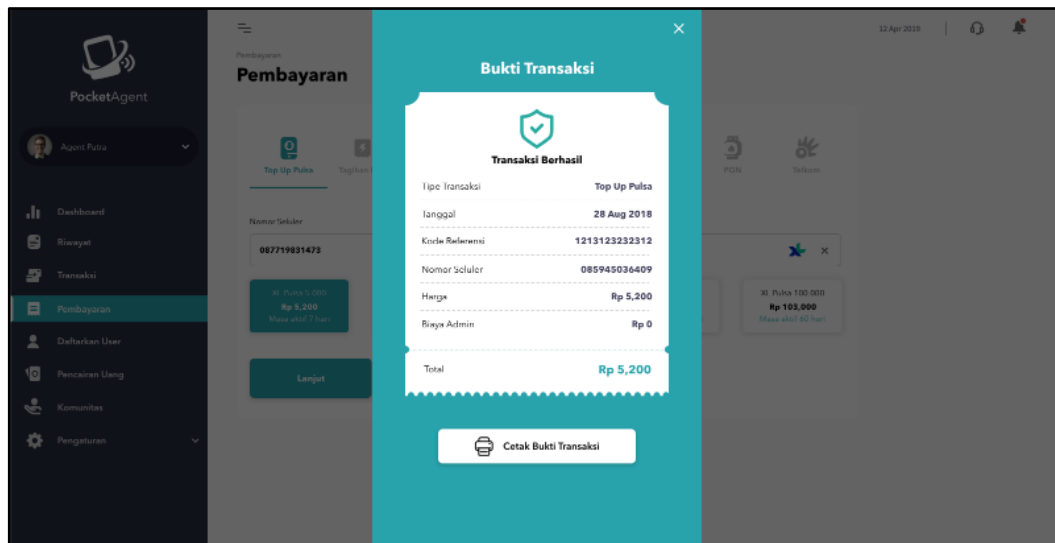
Gambar 3.69 Halaman Mock Up Payment (Lanjutan)

Setelah memilih menu *top up* pulsa yang diinginkan, maka aplikasi akan menampilkan *modal* konfirmasi transaksi untuk memastikan data yang diinput oleh *user*. *User* diminta untuk memastikan nomor ponsel dan jenis menu yang dipilih dengan benar. Setelah memeriksa, maka *user* dapat menekan *button* Benar untuk melanjutkan transaksi. Gambar 3.69 menggambarkan tampilan *modal* untuk melakukan konfirmasi pembelian *m-commerce* khususnya pada *top up* pulsa.



Gambar 3.70 Halaman Mock Up Payment (Lanjutan)

.Setelah menekan *button* Benar, maka *user* akan diminta untuk memasukkan 6 digit PIN Transaksi. Bila *user* lupa PIN Transaksi, maka *user* dapat menekan *link button* Lupa PIN dan *user* akan diarahkan pada halaman Lupa PIN. Setelah memasukkan PIN Transaksi, maka *user* menekan *button* Kirim untuk melanjutkan transaksi pembelian *top up* pulsa. PIN yang dimasukkan secara otomatis diperiksa oleh aplikasi, apabila PIN Transaksi yang dimasukkan salah, maka aplikasi akan mengeluarkan *alert notification* yang menyatakan bahwa PIN Transaksi yang dimasukkan salah. Gambar 3.70 menggambarkan tampilan *modal* konfirmasi pemberian *top up* pulsa.

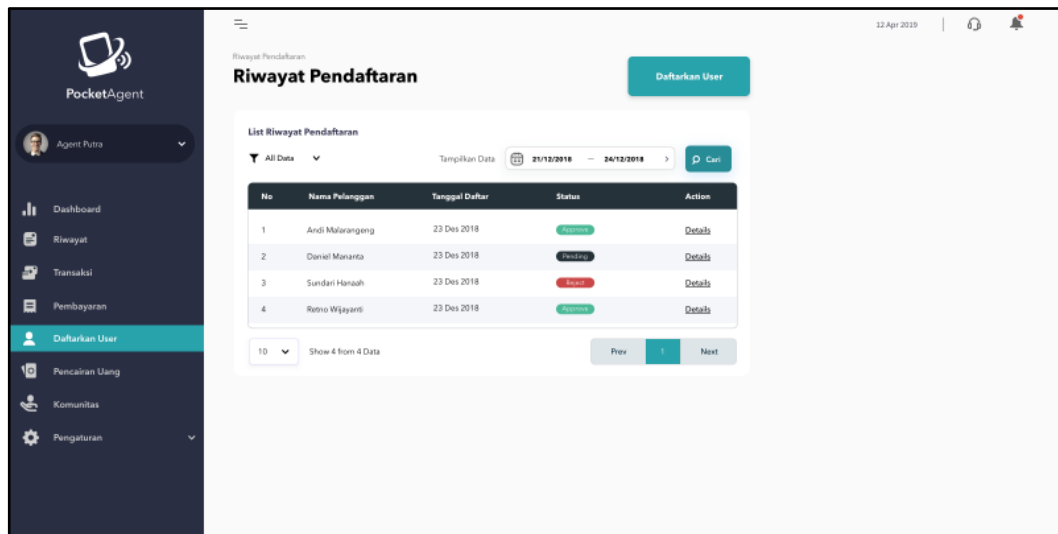


Gambar 3. 71 Halaman Mock Up Payment (Lanjutan)

Setelah selesai memasukkan PIN Transaksi, maka aplikasi akan memberikan bukti transaksi yang menyatakan bahwa bukti pembelian *top up* pulsa telah berhasil dilakukan beserta data waktu pembelian *top up* pulsa. Pada *modal* bukti transaksi, *user* dapat mencetak bukti transaksi dalam bentuk pdf dengan menekan *button* Cetak Bukti Transaksi. Gambar 3.70 menggambarkan bukti transaksi pembelian *m-commerce top up* pulsa.

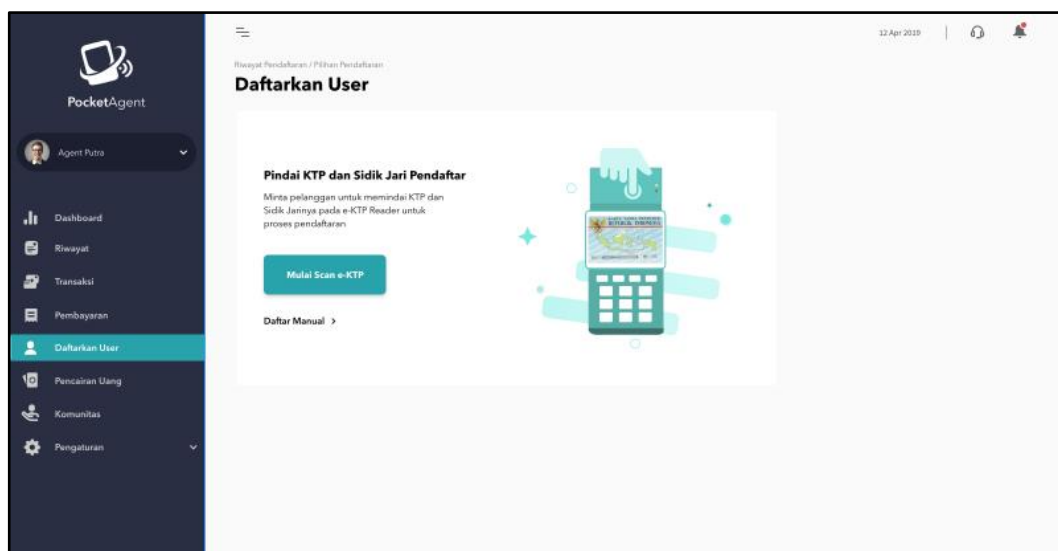
8. Halaman Mock Up Register

Gambar 3.72 sampai dengan Gambar 3.79 menunjukkan rancangan halaman *Payment Page* yang merupakan bagian dari halaman *Registration Page* yang dapat digunakan *user* untuk melakukan registrasi akun untuk member atau *user* lain dengan menggunakan registrasi secara manual atau dengan cara melakukan *scan* e-ktp.



Gambar 3.72 Halaman Mock Up Registration History

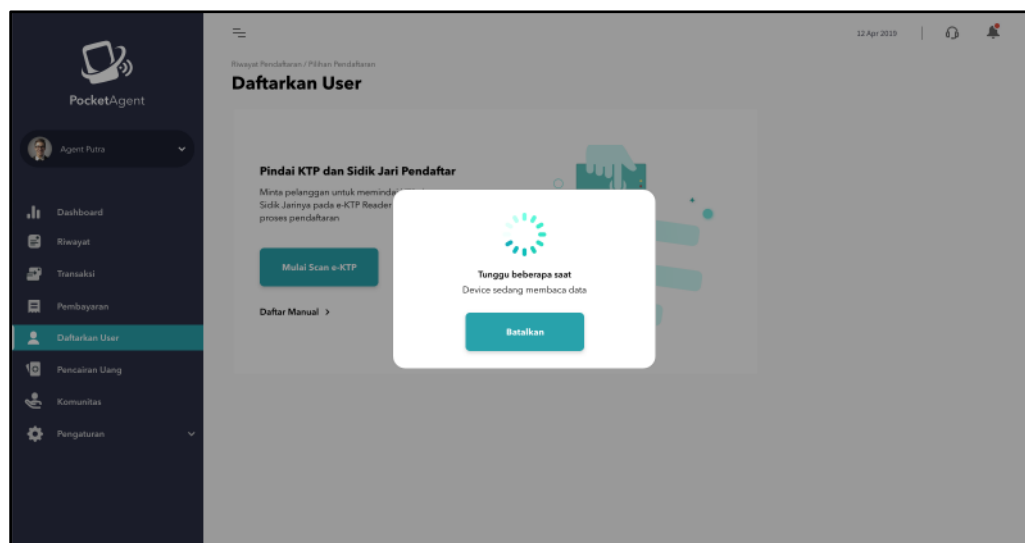
Gambar 3.72 menggambarkan tampilan *list registration history* pada menu Registrasi *user*. Dimana pada halaman ini *user* dapat melihat data *user* lain yang pernah didaftarkan beserta statusnya yang sudah disediakan dalam tabel *registration history*. Pada halaman *Registration History*, *user* dapat melakukan *filter* berdasarkan *range* tanggal tertentu. *User* juga dapat melakukan registrasi untuk orang lain dengan menekan *button* *Daftarkan User*.



Gambar 3.73 Halaman Mock Up Registration

Halaman ini terjadi jika *user* menekan *button* *Daftarkan User* yang ada pada

halaman Gambar 3.72. Pada halaman Daftarkan User, terdapat dua *button*, dimana *user* dapat mendaftarkan orang lain atau menjadi *agent* terhadap orang lain baik secara manual maupun dengan cara melakukan *scan* pada e-KTP. Bila *user* ingin melakukan registrasi pada e-KTP, maka *user* dapat menekan *button* Mulai Scan e-KTP, sedangkan jika *user* ingin mendaftarkan secara manual, maka *user* dapat menekan *link button* Daftar Manual. Gambar 3.73 menggambarkan halaman *registration*.



Gambar 3.74 Halaman Mock Up Registration e-KTP

Gambar 3.74 menggambarkan *loading modal* ketika melakukan *scan* e-KTP. *Loading Modal* dibuat dengan tujuan untuk memastikan bahwa *experince user* terhadap aplikasi masih terjadi yang menandakan bahwa aplikasi sedang membaca data *KTP user* ketika *user* melakukan *scan* e-KTP. *User* dapat menekan *button* Batalkan apabila *user* ingin membatalkan proses registrasi.

PocketAgent

Agent Pura

Dashboard
Riwayat
Transaksi
Pengaturan

Daftarkan User

Data Pribadi

Tipe Identitas *
Kartu Tanda Penduduk

No. Kartu Identitas *
102910019432

Nama Sesuai Identitas *
Ujang Gung

Jenis Kelamin
☒ Laki ☐ Perempuan

Tempat Lahir *
Jakarta

Tanggal Lahir *
10-10-2018

Informasi Kontak

Kota *
Jakarta Selatan

Provinsi *
DKI Jakarta

Alamat *
Jalan Senaya Mulya Puri No. 90

Identitas

No. Handphone *
98128721641

Kode Pos *
152226

Email *
mualif@gmail.com

Upload Dokumen

Upload Foto Kartu Identitas

Upload Foto Customer

Upload Foto selfie dengan Kartu Identitas (Optional)

Daftarkan

Gambar 3.75 Halaman Mock Up Registration e-KTP (Lanjutan)

Ketika aplikasi sudah berhasil membaca data pada e-KTP *user*, maka akan terdapat *form* yang berisi data *user* berdasarkan yang tertulis pada e-KTP beserta foto yang sudah dipisahkan menjadi bagian yang tersendiri. *User* juga masih dapat mengedit data yang dianggap masih tidak sesuai dengan data dirinya dengan cara menekan langsung pada bagian *form* yang ingin diedit. Setelah selesai, *user* dapat menekan *button* Daftarkan untuk melanjutkan proses registrasi. Gambar 3.75 menggambarkan halaman jika registrasi *e-ktp* terbaca oleh aplikasi.

PocketAgent

Agent Prita

Dashboard
Riwayat
Transaksi
Pembayaran
Daftarkan User
Pencarian Uang
Kembalian
Pengaturan

12 Apr 2019

Daftarkan User

Formulir Pendaftaran / Pilihan Pendaftaran / Daftarkan User

Data Diri

Tipe Identitas *

Kartu Tanda Penduduk

No. Kartu Identitas *

Ccontoh: 201901821514

Nama Sesuai Identitas *

Ccontoh: 201901821514

Jenis Kelamin

☒ Male ☐ Female

Tanggal Lahir *

Ccontoh: 201901821514

Informasi Alamat

Provinsi *

DKI Jakarta

Kota *

Alamat *

Lengkapi Profil

No. Handphone *

Email *

Kode Pos (Sesuai Alamat Identitas) *

Upload Dokumen

Upload Foto Kartu Identitas

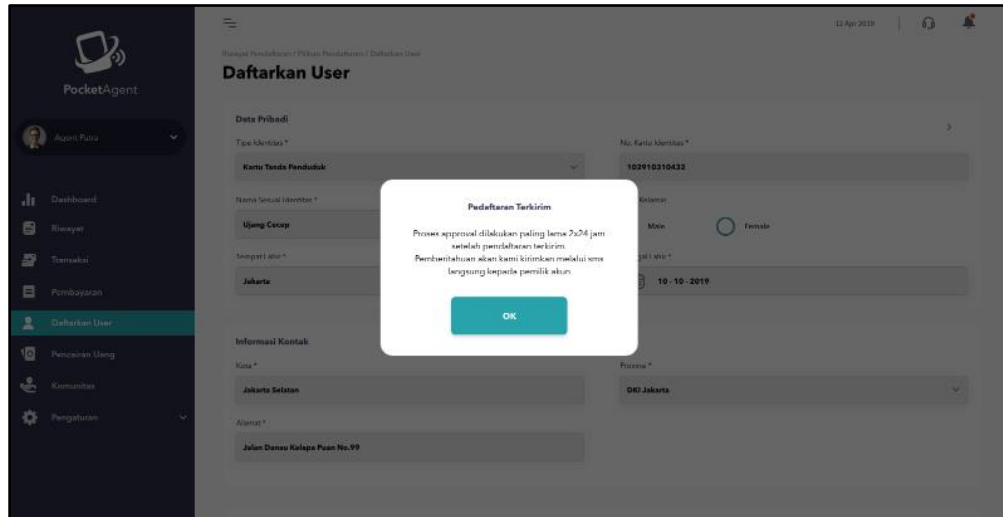
Upload Foto selfie dengan Kartu Identitas (Optional)

Upload Foto Customer

Daftarkan

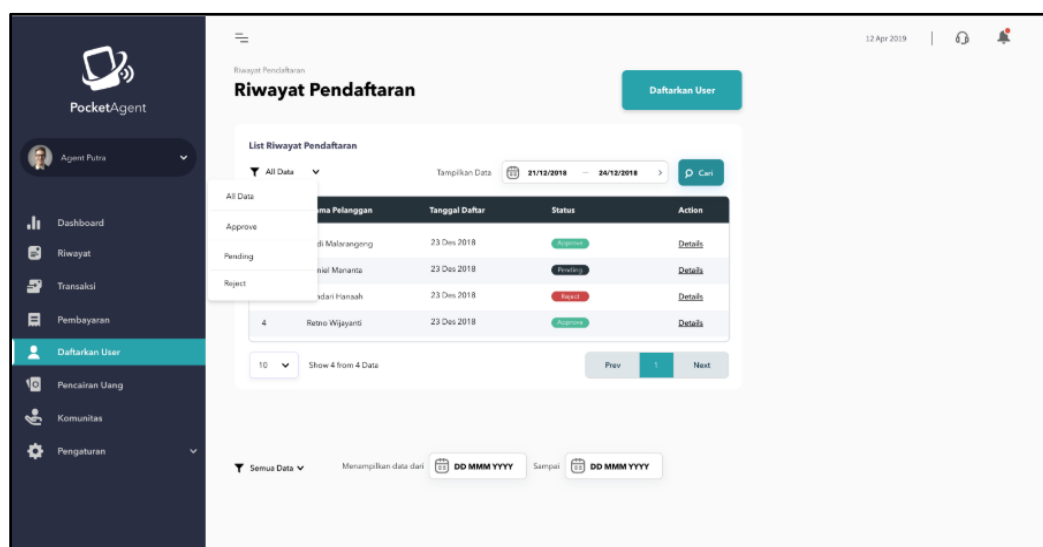
Gambar 3.76 Halaman Mock Up Registration manual

Gambar 3.76 menggambarkan tampilan *form* registrasi manual. Pada halaman ini terdapat *form registration* yang masih kosong. Pada halaman ini, *user* diminta untuk mengisi semua *form* sebagai syarat data untuk memenuhi kebutuhan registrasi seperti data pada e-KTP dan foto. Setelah mengisi data secara manual *user* dapat menekan *button* Daftarkan.



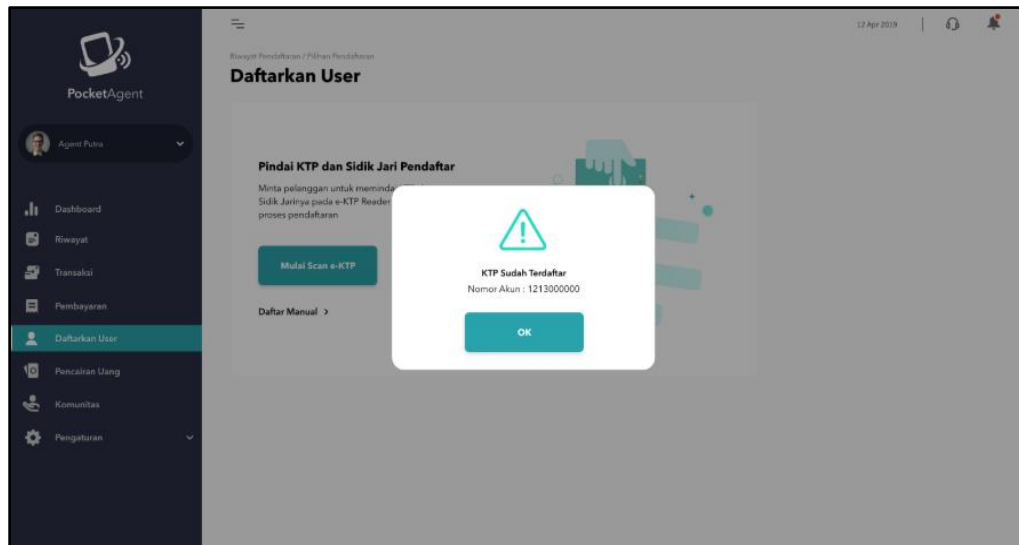
Gambar 3.77 Halaman Mock Up Registration (Lanjutan)

Setelah *user* menekan *button* Daftarkan, maka aplikasi akan memeriksa dan memastikan semua data yang diperlukan sudah terinput dengan baik sesuai format pengisian *form* yang sesuai. Setelah semua data diinput dengan benar, maka aplikasi akan mengeluarkan *modal* yang menandakan *registration success* yang mengatakan bahwa proses registrasi akan diproses lebih lanjut jika proses registrasi selesai, maka *user* akan mendapat *message* terkait dengan status proses registrasi Gambar 3.77 menggambarkan ketika proses registrasi berhasil.



Gambar 3.78 Halaman Mock Up Registration History Filter

Pada halaman *Registration History*, *user* dapat melakukan *filter* berdasarkan jenis status dan berdasarkan *range* tanggal tertentu. Fungsi *filter* dibutuhkan untuk mempermudah pemeriksaan status terhadap status proses registrasi *user* yang dilakukan sebelumnya. Gambar 3.78 menggambarkan *filter* pada tabel *registration history*.



Gambar 3.79 Halaman Mock Up Registration E-KTP Scan Failed

Gambar 3.79 menggambarkan *loading modal* ketika *e-ktp* tidak terbaca oleh *scanner*. Dimana pada halaman ini terdiri dari *loading modal* yang berisi *icon warning* yang menandakan bahwa *e-KTP* yang di *scan* tidak terbaca oleh *user*. *User* dapat menekan *button* OK dan menekan kembali *button* Daftarkan User seperti pada Gambar 3.73 dan melakukan proses *scan* ulang *e-KTP* untuk melanjutkan proses registrasi.

9. Halaman *Mock Up Redeem Money*

Gambar 3.80 sampai dengan Gambar 3.86 menunjukkan rancangan halaman *Redeem Money* yang dapat digunakan *user* untuk melakukan pencairan uang melalui sesama *agent group* atau dengan *account* lain dan pada halaman ini, *user* dapat melihat seluruh transaksi pencairan uang yang pernah dilakukan

sebelumnya.

Tanggal	Akun Asal	Akun Tujuan	Nominal	Status	Aksi
23 Des 2018	211910390101	8171321983110	50.000	Lulus	Detail
23 Des 2018	211910390101	8171321983110	5.000.000	Lulus	Detail
23 Des 2018	211910390101	8171321983110	50.000	Lulus	Detail
23 Des 2018	211910390101	8171321983110	20.000.000	Lulus	Detail

Gambar 3.80 Halaman Mock Up Redeem Money

Gambar 3.80 menggambarkan tampilan halaman *redeem money*. Dimana pada halaman *redeem money* terdapat dua bagian yaitu bagian *form* untuk pencairan uang dan bagian tabel yang berisi riwayat transaksi yang pernah terjadi sebelumnya. Pada bagian *form* untuk pencairan uang, dapat digunakan oleh *user* untuk mencairkan uang dari hasil komisi yang dilakukan sebagai *agent*, sedangkan pada bagian tabel riwayat transaksi, *user* dapat melihat semua riwayat komisi yang pernah didapatkan beserta tanggal terjadinya transaksi. Pada bagian tabel riwayat transaksi, *user* dapat melakukan filter dan *search* untuk mempermudah pencarian riwayat transaksi.

PocketAgent

Agent Putra

- Dashboard
- Riwayat
- Transaksi
- Pembayaran
- Registrasi Pelanggan
- Pencairan Uang**
- Komunitas
- Pengaturan

Pencairan Uang / Form Pencairan

Pencairan Uang

Kode Referensi*
67766777765

Akun Asal*
211910390101

Akun Tujuan*
8171371983110

Lanjut

Riwayat Transaksi

Filter >

Tanggal	Akun Asal	Akun Tujuan	Nominal	Status	Aksi
23 Des 2018	211910390101	8171371983110	50,000	Saldo	Detail
23 Des 2018	211910390101	8171371983110	5.000.000	Saldo	Detail
23 Des 2018	211910390101	8171371983110	50,000	Saldo	Detail
23 Des 2018	211910390101	8171371983110	20.000.000	Saldo	Detail

10 Show 4 from 4 Data

Prev 1 Next

Gambar 3.81 Halaman Mock Up Redeem Money (Lanjutan)

Pada bagian *grid redeem money*, *user* diminta untuk memasukkan kode referensi dan nomor akun asal dan nomor akun tujuan untuk melakukan proses *redeem money*. Setelah selesai mengisi semua *form*, maka *user* dapat menekan *button* Lanjut dan secara otomatis aplikasi akan mendeteksi dan memvalidasi *user input* yang sudah diisi *user* sebelumnya. Gambar 3.81 menggambarkan tampilan ketika *form reedem money* diisi.

PocketAgent

Agent Putra

- Dashboard
- Riwayat
- Transaksi
- Pembayaran
- Registrasi Pelanggan
- Pencairan Uang**
- Komunitas
- Pengaturan

Pencairan Uang / Form Pencairan

Pencairan Uang

Kode Referensi*
67766777765

Akun Asal*
211910390101

Akun Tujuan*
8171371983110

Lanjut

Riwayat Transaksi

Filter >

Konfirmasi

Pencairan Uang

Nama: Andi

Kode Referensi: 091283141091

Akun Asal: 211910390101

Akun Tujuan: 8171371983110

Nominal Pencairan: Rp 50,000

Biaya Admin: -

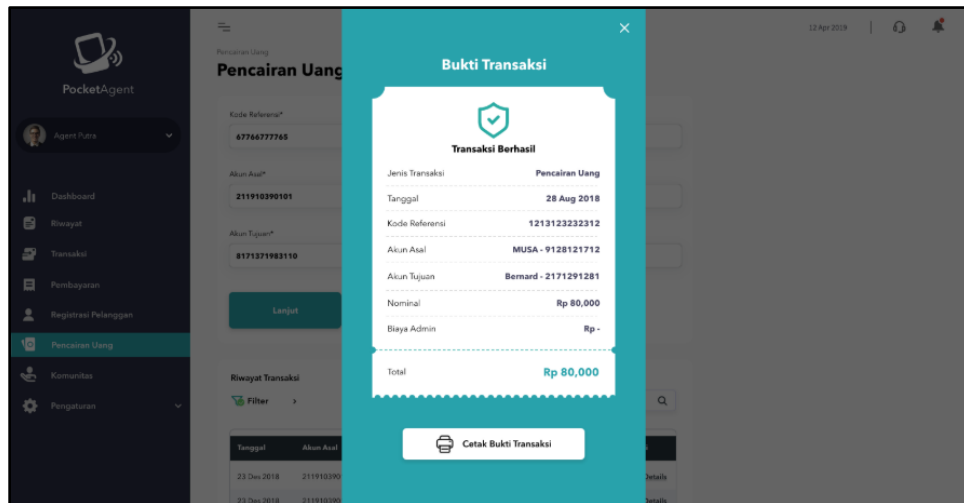
Total Bayar: **Rp80,000**

Benar

Gambar 3.82 Halaman Mock Up Redeem Money (Lanjutan)

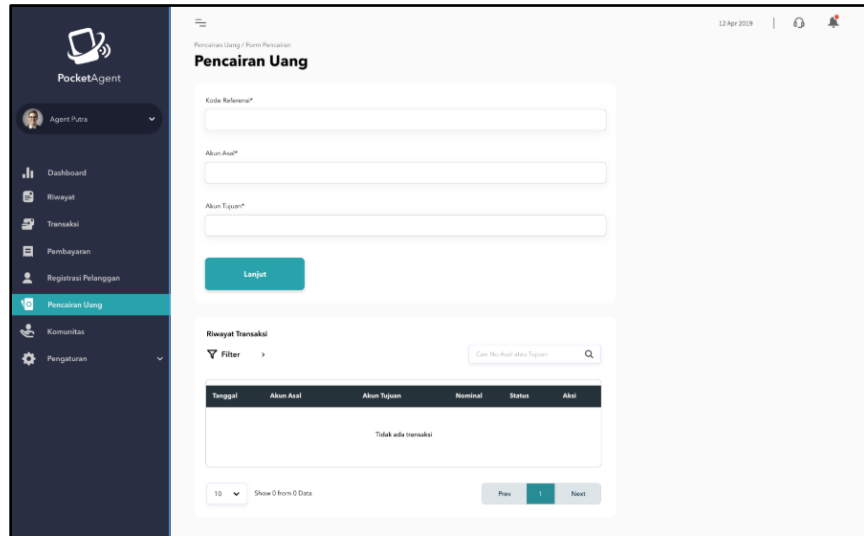
Setelah *user* menekan *button* lanjut, maka *user* akan mendapatkan *modal confirmation* untuk melakukan konfirmasi ulang tentang data *input* yang

sebelumnya *user* lakukan. Pada *modal* ini *user* diminta untuk memeriksa kembali terkait dengan nama dan nomor akun yang diinput sebelumnya untuk memastikan bahwa *data input* yang dimasukkan benar. Setelah memastikan data dengan benar, maka *user* dapat melanjutkan proses transaksi dengan menekan *button* Benar. Gambar 3.82 menggambarkan *modal* konfirmasi untuk melakukan pencairan uang.



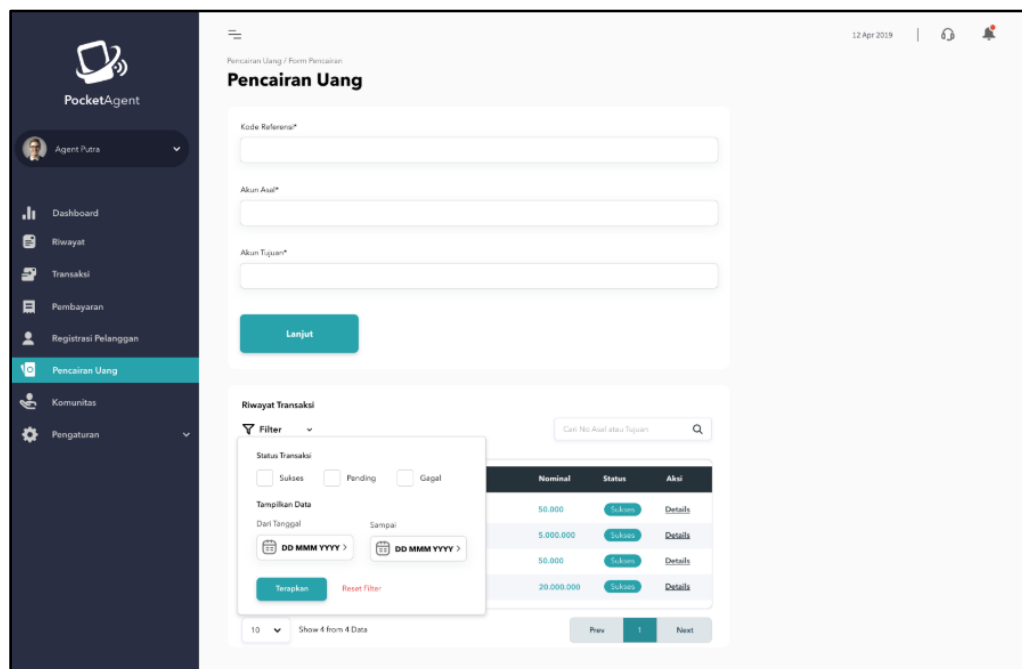
Gambar 3.83 Halaman Mock Up Redeem Money (Lanjutan)

Setelah *user* menekan *button* Benar pada *modal* sebelumnya, maka *user* mendapatkan bukti transaksi yang menandakan bahwa transaksi pencairan uang yang dilakukan telah berhasil beserta data transaksi yang dilakukan. Pada *modal* halaman ini, *user* dapat mencetak Bukti transaksi dalam bentuk pdf dengan menekan *button* Cetak Bukti Transaksi. Gambar 3.83 menggambarkan tampilan bukti sukses pencairan uang beserta data transaksi.



Gambar 3.84 Halaman Mock Up Redeem Money (Lanjutan)

Gambar 3.84 menggambarkan tampilan ketika tidak pernah terjadi pencairan uang. Dimana pada halaman ini *user* belum pernah melakukan transaksi pencairan uang sehingga belum terdapat data pada tabel riwayat transaksi. Namun *user* tetap dapat melakukan pencairan uang terhadap aplikasi dengan mengisi *form redeem money* pada *grid* bagian atas tabel riwayat transaksi.

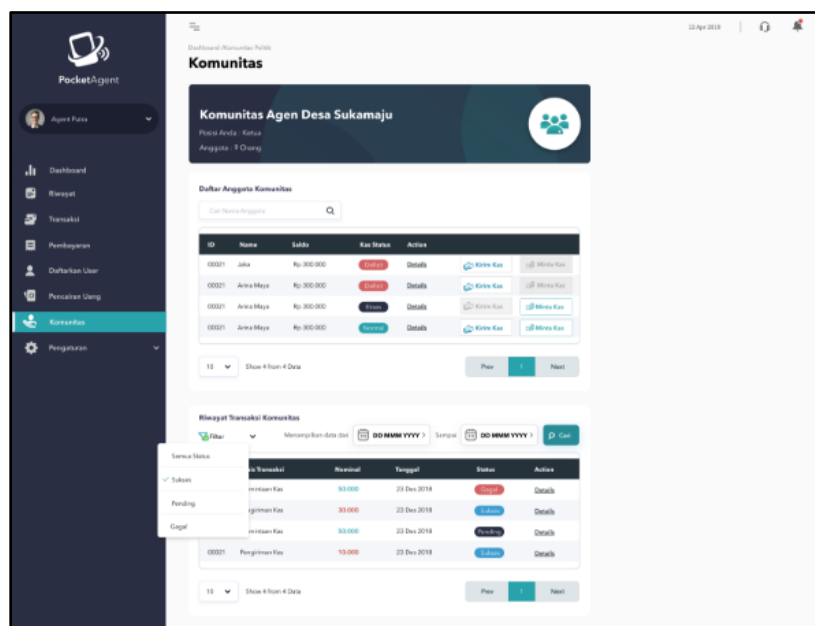


Gambar 3.85 Halaman Mock Up Reedem Money Filter

Pada bagian tabel riwayat transaksi, *user* dapat melakukan *filter* yang terdiri dari beberapa macam jenis *filter* yang dapat dipilih berdasarkan status transaksi. *User* dapat mencari riwayat transaksi berdasarkan status transaksi gagal, berhasil, atau *pending* lalu *user* juga dapat melakukan *filter* terhadap *range date* tertentu untuk mempermudah pencarian riwayat transaksi lalu *user* dapat menekan *button* OK dan secara otomatis tabel akan menampilkan data berdasarkan kriteria yang dipilih *user*. Bila *user* juga dapat melakukan *reset* dengan menekan *link button* Reset untuk menghapus semua *filter* agar kembali seperti semula. Gambar 3.85 menggambarkan *filter* pada fitur pencairan uang.

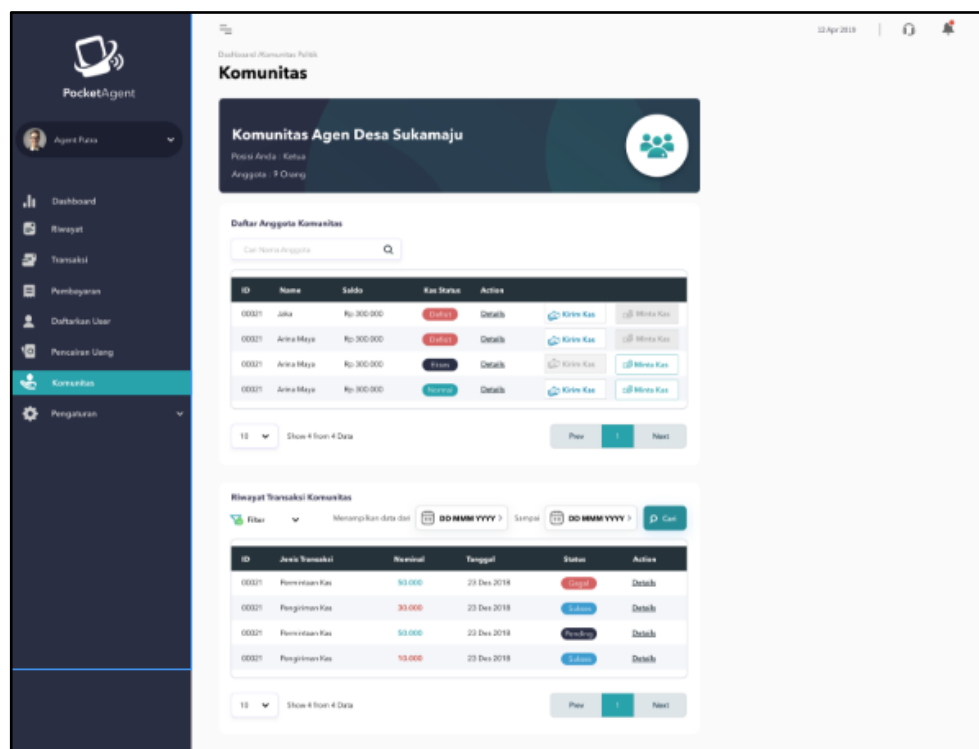
10. Halaman *Mock Up Community*

Gambar 3.86 sampai dengan Gambar 3.102 menunjukkan rancangan halaman *Community* yang dibuat ketika *user* mendaftarkan orang lain. Fitur ini dapat digunakan untuk meminta uang atau mengirim uang, *user* juga dapat melakukan filter sesuai status untuk melihat transaksi yang pernah dilakukan.

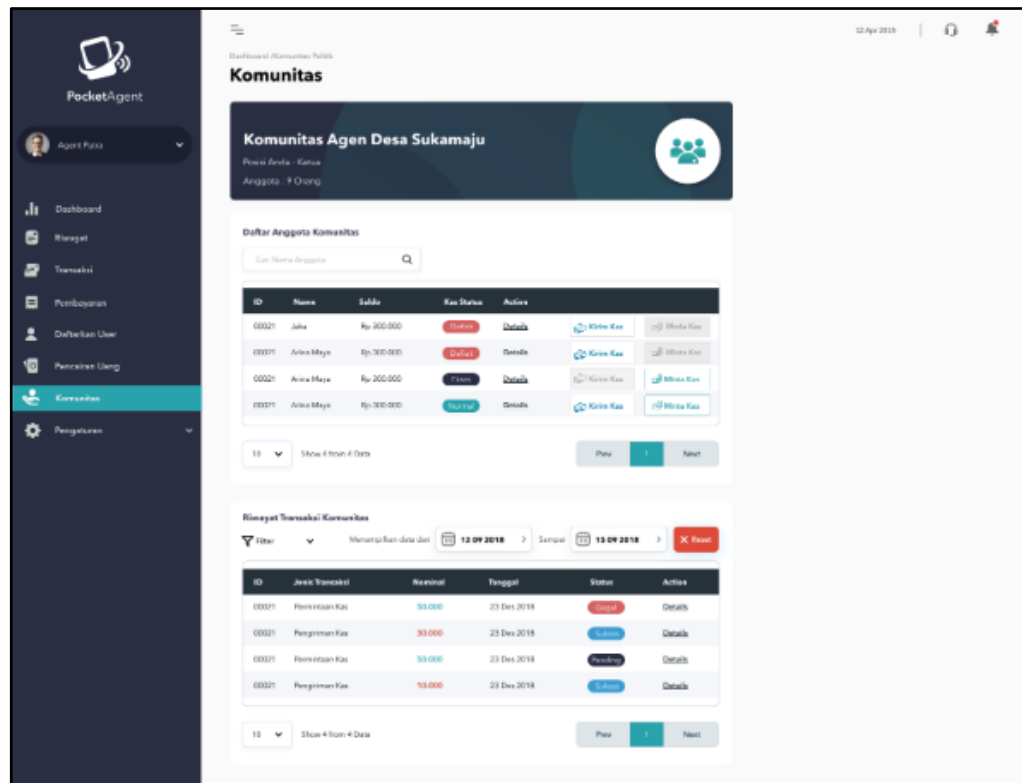


Gambar 3.86 Halaman Mock Up Community

Pada Halaman *Community*, *user* dapat membuat komunitas sesuai dengan member yang pernah didaftarkan. Pada halaman ini, terdapat dua bagian tabel. Pada tabel yang pertama yang berada di bawah *breadcrumb* judul halaman, *user* dapat melihat list *user* lain yang menjadi komunitasnya. *User* juga dapat melihat status *member community* dan meminta serta mengirim uang kas kepada *member* tersebut. Pada tabel *list community*, *user* juga dapat melakukan *search* untuk mencari nama *member* agar lebih mudah. Sedangkan pada tabel yang berada dibawah tabel *list community*, *user* dapat melihat riwayat transaksi antar *member* yang berada di dalam *community* beserta dengan status transaksi yang dilakukan. *User* juga dapat melakukan *filter* berdasarkan status dan *range date* tertentu untuk mempermudah pencarian riwayat transaksi yang pernah dilakukan. Gambar 3.86 menggambarkan *filter* pada *community* dan Gambar 3.87 menggambarkan tampilan Halaman *Community* secara keseluruhan.

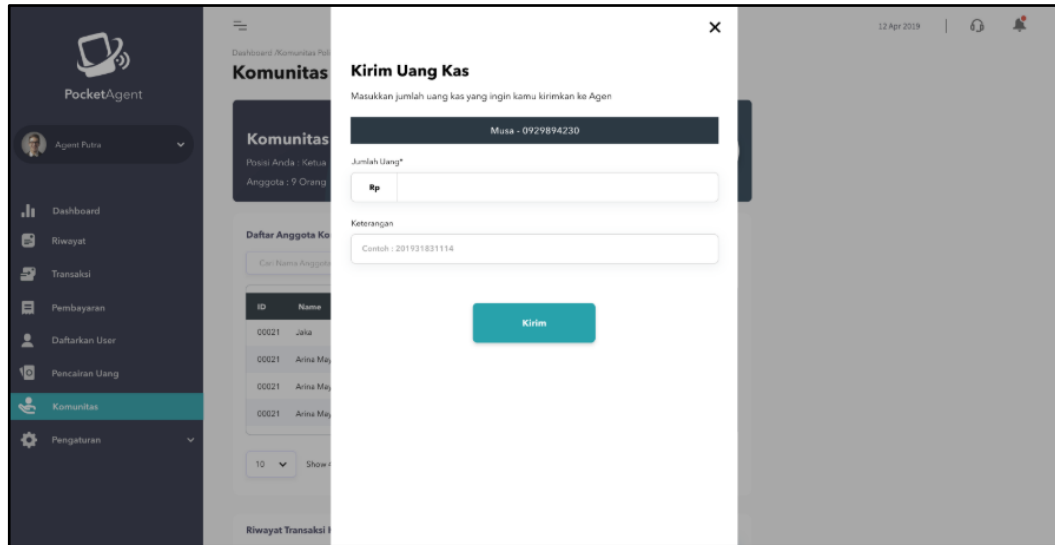


Gambar 3.87 Halaman Mock Up Comunity (Lanjutan)



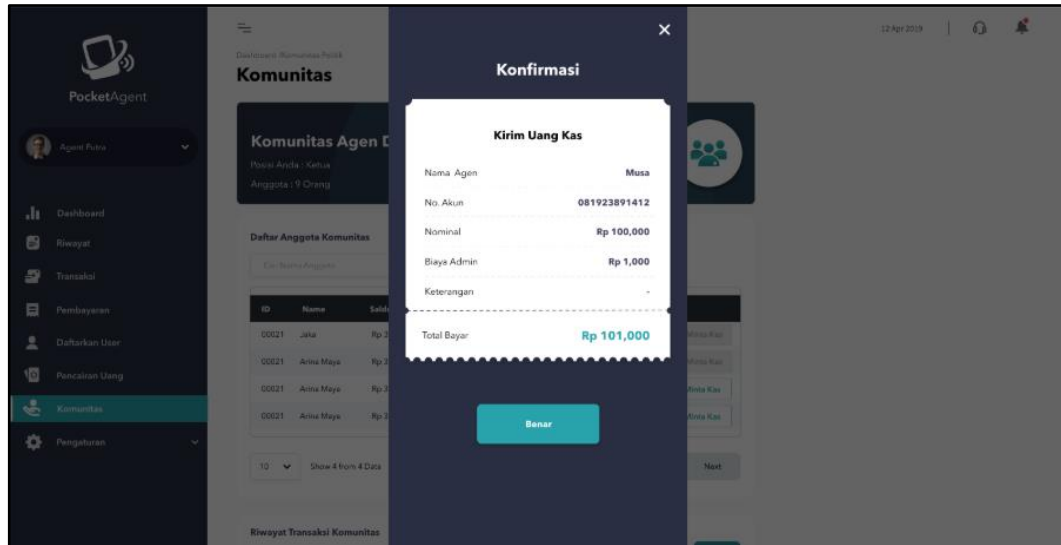
Gambar 3.88 Halaman Mock Up Comunity (Lanjutan)

Jika *user* sudah melakukan *filter* maka tabel riwayat transaksi pada komunitas akan menampilkan data sesuai dengan *filter* yang dilakukan *user* sebelumnya. *User* juga dapat melakukan *reset* untuk menampilkan data pada tabel seperti keadaan semula dengan menekan *button reset* yang berada di sebelah *dropdown range date*. Maka data pada tabel riwayat transaksi *community* akan kembali seperti semula. Gambar 3.88 menggambarkan untuk melakukan *reset* pada *filter*.



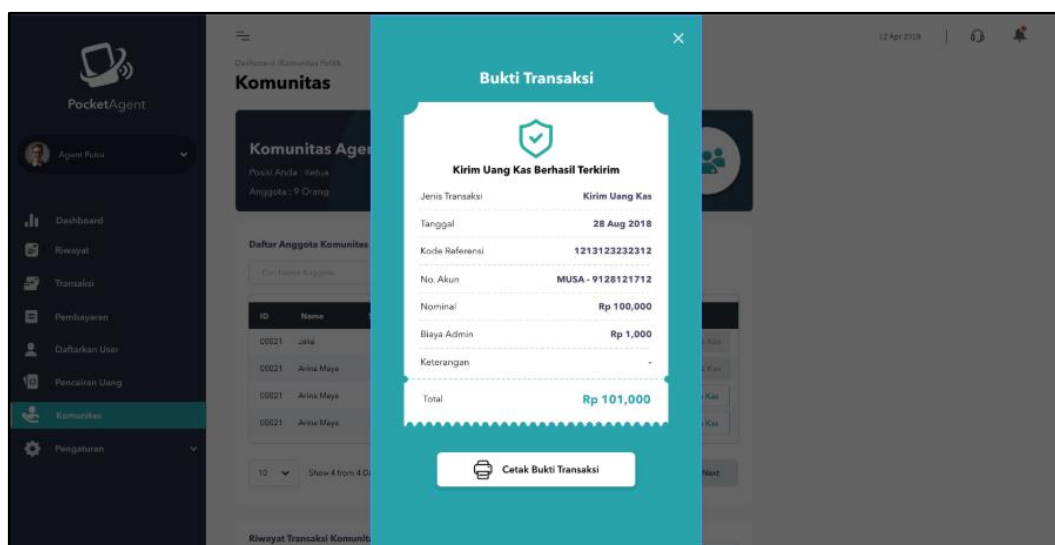
Gambar 3.89 Halaman Mock Up Comunity (Lanjutan)

Jika *user* menekan *button* pada Kirim Uang Kas pada nama *member* tertentu dalam tabel *list member community*, maka aplikasi akan mengeluarkan *modal* Kirim Uang Kas. Pada *modal* Kirim Uang Kas terdapat nama dan nomor rekening *member* yang ingin dikirimkan uang kas. Lalu *user* juga diminta untuk memasukkan jumlah uang berdasarkan nominal uang yang ingin diberikan lalu bila perlu *user* dapat memasukkan keterangan transaksi. Setelah memasukkan semua data yang dibutuhkan maka *user* dapat menekan *button* Kirim untuk melanjutkan transaksi pengiriman uang terhadap *member* tersebut. Gambar 3.89 menggambarkan *modal* untuk mengirimkan uang kas kepada sesama komunitas.



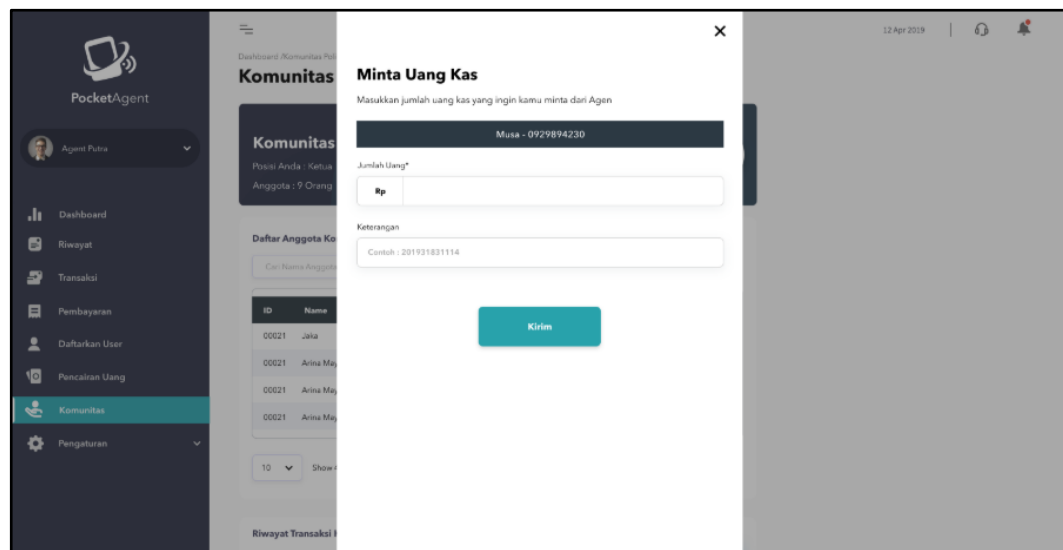
Gambar 3.90 Halaman Mock Up Comunity (Lanjutan)

Setelah *user* menekan *button* Kirim, maka *user* akan diminta untuk melakukan konfirmasi terhadap data yang dimasukkan sebelumnya dengan mengeluarkan *modal* untuk melakukan pemeriksaan kembali terhadap data *input*. Setelah *user* memastikan data pada *modal confirmation* benar, maka *user* dapat menekan *button* Benar untuk melanjutkan transaksi kirim uang kas. Gambar 3.90 menggambarkan modal konfirmasi untuk melakukan pemeriksaan ulang terhadap *data input user*.



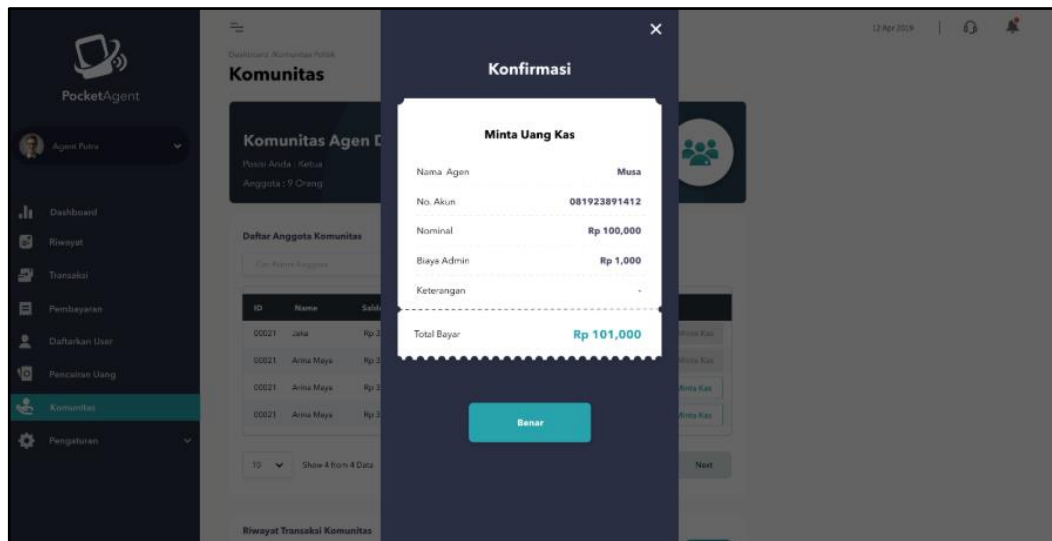
Gambar 3.91 Halaman Mock Up Comunity (Lanjutan)

Setelah *user* menyetujui halaman sebelumnya pada Gambar 3.90, maka aplikasi akan mengeluarkan *modal* bukti transaksi yang menyatakan bahwa transaksi yang dilakukan sukses dilakukan. *User* dapat mencetak transaksi dalam bentuk pdf sebagai bukti transaksi yang sah dengan menekan *button* Cetak Hasil Transaksi. Gambar 3.91 menggambarkan bukti transaksi mengirimkan uang kas kepada komunitas.



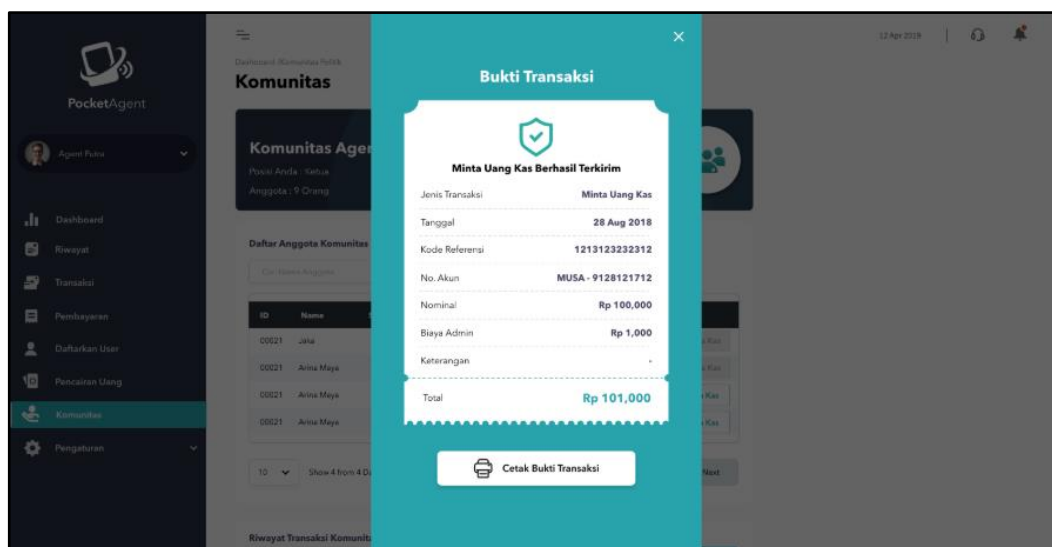
Gambar 3.92 Halaman Mock Up Comunity (Lanjutan)

User juga dapat meminta uang kas pada salah satu *list member* yang ada pada tabel *list member community* dengan menekan *Button* Minta Uang Kas, lalu aplikasi akan mengeluarkan *modal* minta uang kas yang berisikan nama dan nomor rekening *member* yang akan diminta uang kasnya, lalu *user* diminta untuk memasukkan jumlah uang yang diinginkan serta bila diperlukan, *user* dapat menambahkan keterangan. Setelah selesai, *user* dapat menekan *button* Kirim untuk melanjutkan transaksi Minta Kas. Gambar 3.92 menggambarkan *modal* minta uang kas kepada sesama komunitas.



Gambar 3.93 Halaman Mock Up Community (Lanjutan)

Setelah *user* menekan *button* Kirim pada Gambar 3.92, *user* akan diminta untuk melakukan konfirmasi mengenai data yang dimasukkan sebelumnya untuk memastikan data yang dimasukkan *user* telah benar. Setelah *user* setuju dengan data yang muncul pada *modal* konfirmasi, maka *user* dapat menekan *button* Benar untuk melanjutkan transaksi. Gambar 3.93 menggambarkan *modal* konfirmasi untuk meminta uang kas kepada sesama anggota komunitas.



Gambar 3.94 Halaman Mock Up Community (Lanjutan)

Setelah *user* menekan *button* Benar, maka *user* akan mendapatkan *modal*

Bukti Transaksi beserta data transaksi yang menyatakan bahwa transaksi yang dilakukan sukses dan *user* dapat mencetak bukti transaksi tersebut dengan menekan *button* Cetak Bukti Transaksi yang ada pada bagian bawah *modal* sebagai bukti transaksi yang sah. Gambar 3.94 menggambarkan bukti transaksi meminta uang kas kepada sesama anggota komunitas.

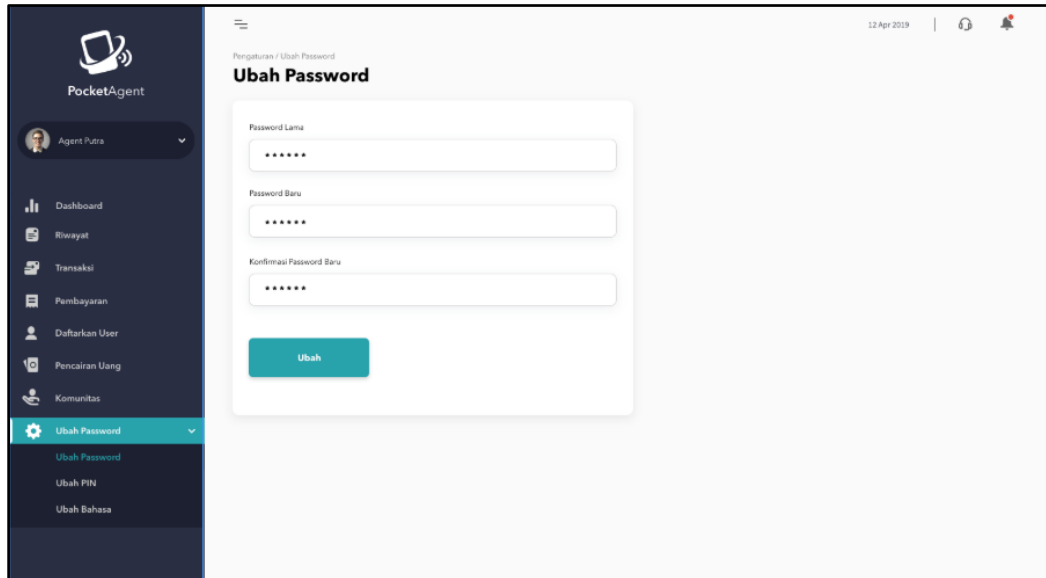
11. Halaman *Mock Up Setting*

11.1. Halaman *Mock Up Change Password*

Gambar 3.95 sampai dengan Gambar 3.97 menunjukkan rancangan halaman *Change Password* yang dibuat ketika *user* mengganti *password* pada aplikasi dengan memasukkan *password* lama dengan *password* baru.

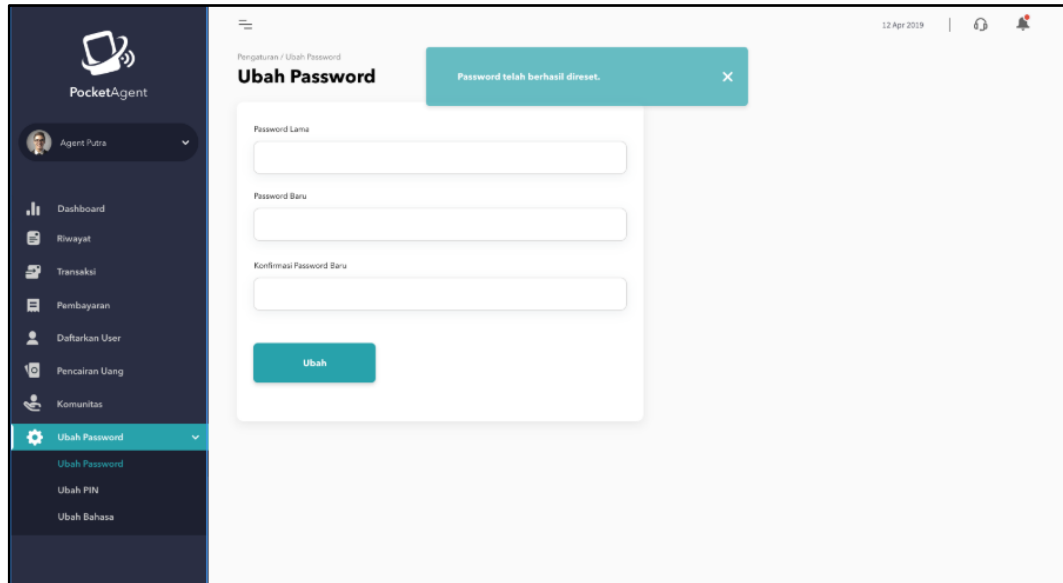
Gambar 3.95 Halaman Change Password

Gambar 3.95 menggambarkan tampilan halaman *change password* pada menu *settings*. Pada halaman ini terdiri dari *form* untuk mengubah *password* Aplikasi Pocketbank Agent. *Form* terdiri dari *password* lama, *password* baru, dan ketik ulang *password* baru yang harus diisi *user* untuk mengganti *password* pada aplikasi. Setelah selesai, maka *user* dapat menekan *button* Ubah untuk mengubah *password*.



Gambar 3.96 Halaman Change Password (Lanjutan)

Gambar 3.96 menggambarkan tampilan ketika *form change password* telah diisi. Dimana pada semua *form* ini, wajib diisi oleh *user* untuk mengganti *password*. *Password* yang diisi oleh *user* wajib terdiri dari minimal terdiri dari 1 huruf kapital, terdapat minimal terdapat 1 angka dan memiliki karakter lebih dari 6 karakter. Jika *user* salah dalam memasukkan *password* maka secara otomatis *form* akan memunculkan validasi seperti *invalid format*. *Password* baru yang akan digunakan tidak boleh menggunakan *password* lama dan *password* lama yang sudah pernah dipakai tidak dapat digunakan sementara dalam satu kali pergantian *password* setelahnya. Setelah selesai *user* dapat menekan *button* Ubah untuk mengubah *password*.

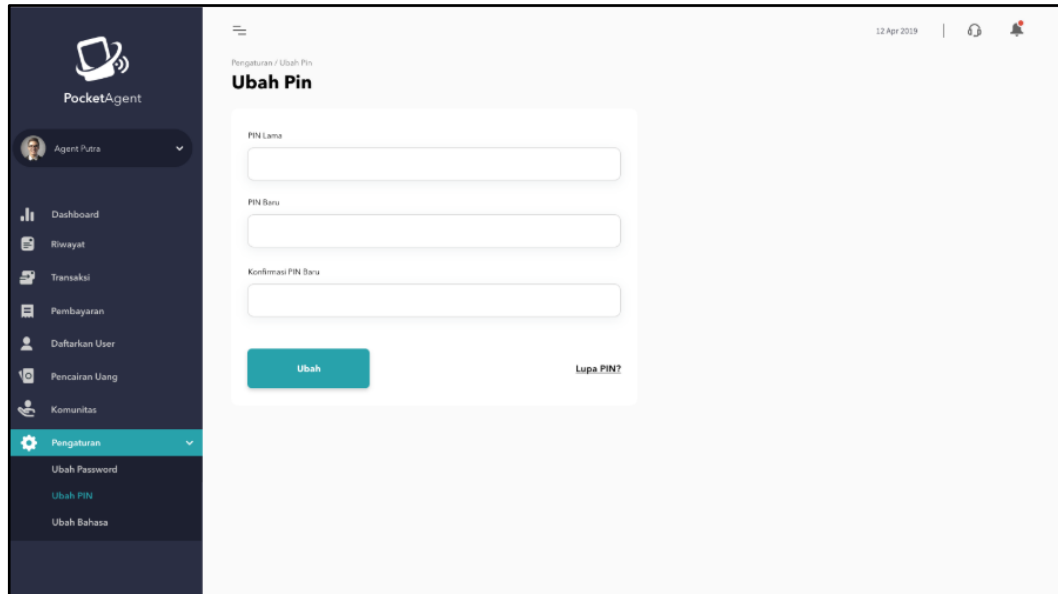


Gambar 3. 97 Halaman Change Password (Lanjutan)

Gambar 3.97 menggambarkan *notification* ketika *change password* berhasil dilakukan maka aplikasi akan mengeluarkan notifikasi *password* sudah berhasil diganti dan secara otomatis, aplikasi langsung *logout*. Kemudian *user* diminta untuk *login* dengan menggunakan *password* yang baru.

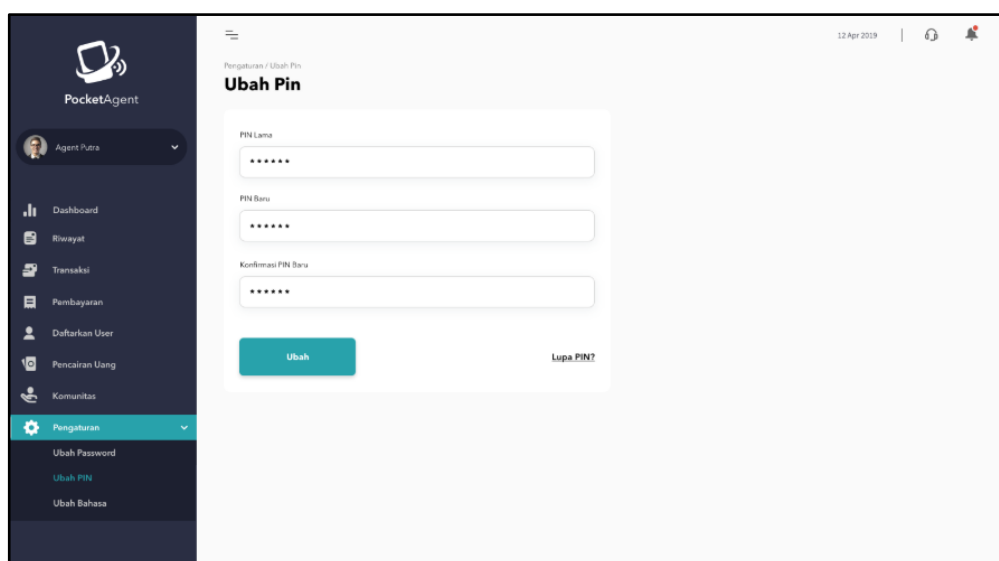
11.2. Halaman *Change PIN*

Gambar 3.98 sampai dengan Gambar 3.100 menunjukkan rancangan halaman *Change PIN* yang dibuat ketika *user* mengganti PIN pada aplikasi dengan memasukkan PIN lama dengan PIN baru.



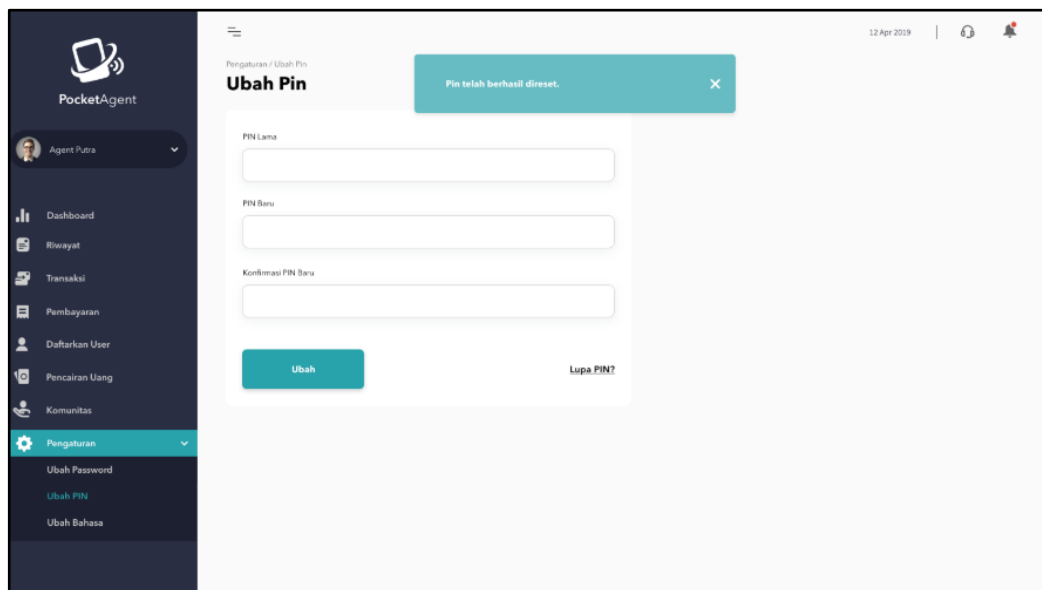
Gambar 3.98 Halaman Mock Up Change PIN

Gambar 3.98 menggambarkan halaman *change* PIN. Pada halaman ini *user* dapat mengganti PIN transaksi dengan memasukkan PIN Lama serta memasukkan PIN baru dan *user* wajib untuk mengetik kembali PIN barunya untuk melakukan validasi kesamaan PIN yang baru diganti. Setelah itu *user* dapat menekan *button* Kirim untuk melanjutkan proses ubah PIN. Bila *user* lupa PIN transaksi, maka *user* dapat menekan *button* Lupa PIN untuk mereset ulang PIN transaksi yang dimiliki.



Gambar 3. 99 Halaman Mock Up Change PIN (Lanjutan)

Gambar 3.99 menggambarkan halaman *form change* PIN yang sudah diisi. PIN transaksi terdiri dari 6 angka. Ketika *user* memasukkan data pada *form* Ubah PIN dan menekan *button* Kirim, maka secara otomatis aplikasi akan mendeteksi kebenaran PIN lama yang dimasukkan oleh *user*. Bila PIN lama yang dimasukkan *user* salah, maka aplikasi akan mengeluarkan notifikasi bahwa PIN lama yang dimasukkan salah..



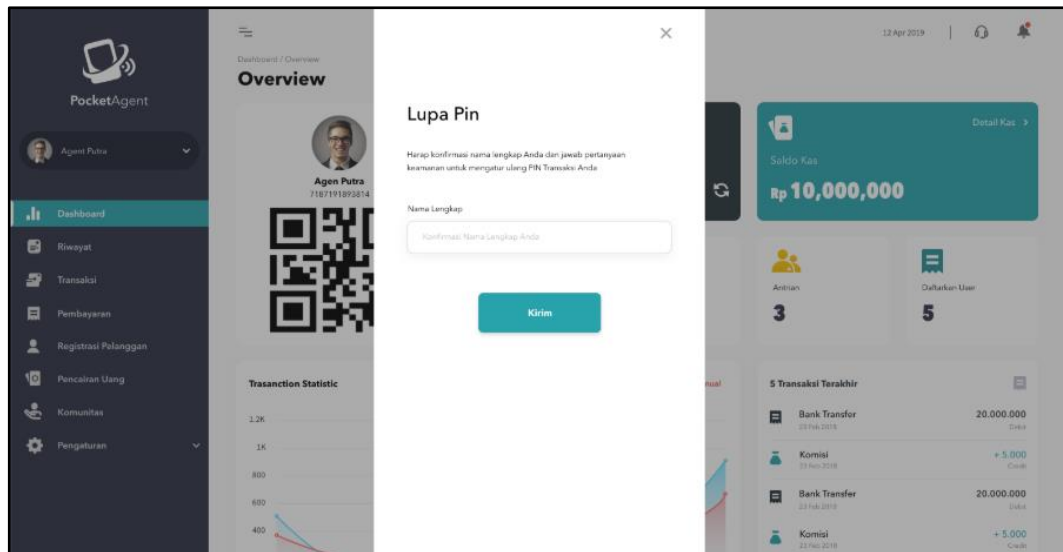
Gambar 3.100 Halaman Mock Up Change PIN

Setelah *user* sukses mengubah PIN, maka aplikasi akan mengeluarkan notifikasi yang menyatakan bahwa PIN transaksi sudah berhasil diganti dan *user* dapat langsung menggunakan PIN yang baru untuk melakukan transaksi selanjutnya. Gambar 3.100 menggambarkan *notification* ketika proses ubah PIN berhasil.

11.3. Halaman Forgot PIN

Gambar 3.101 sampai dengan Gambar 3.106 menunjukkan rancangan halaman *Forgot* PIN yang dibuat ketika *user* jika lupa PIN pada aplikasi dengan melakukan validasi dan verifikasi nomor telepon yang didaftarkan dan

memasukkan kode *One Time Password* (OTP) lalu menjawab pertanyaan keamanan yang pernah dijawab ketika *activation account*.

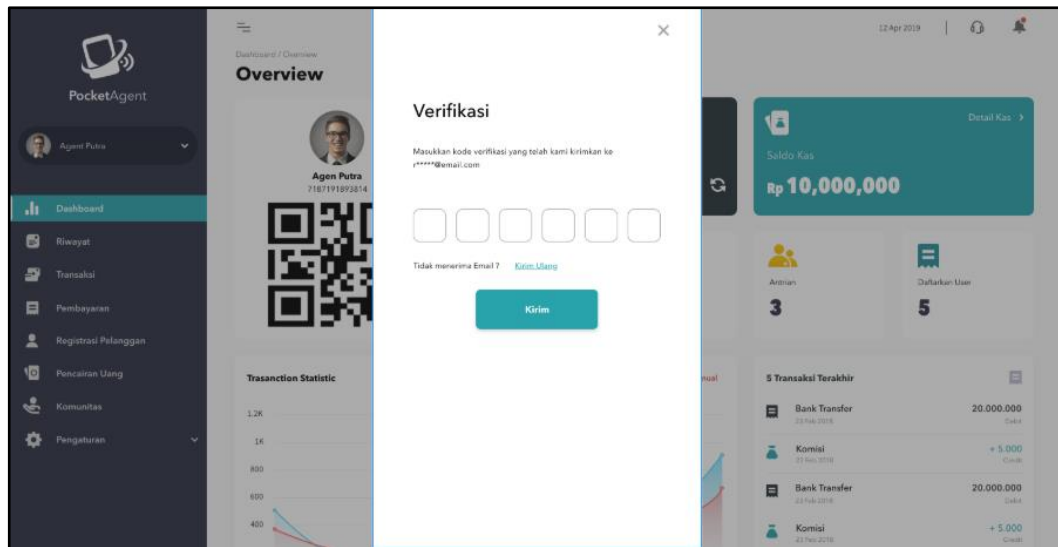


Gambar 3.101 Halaman Mock Up Forgot PIN

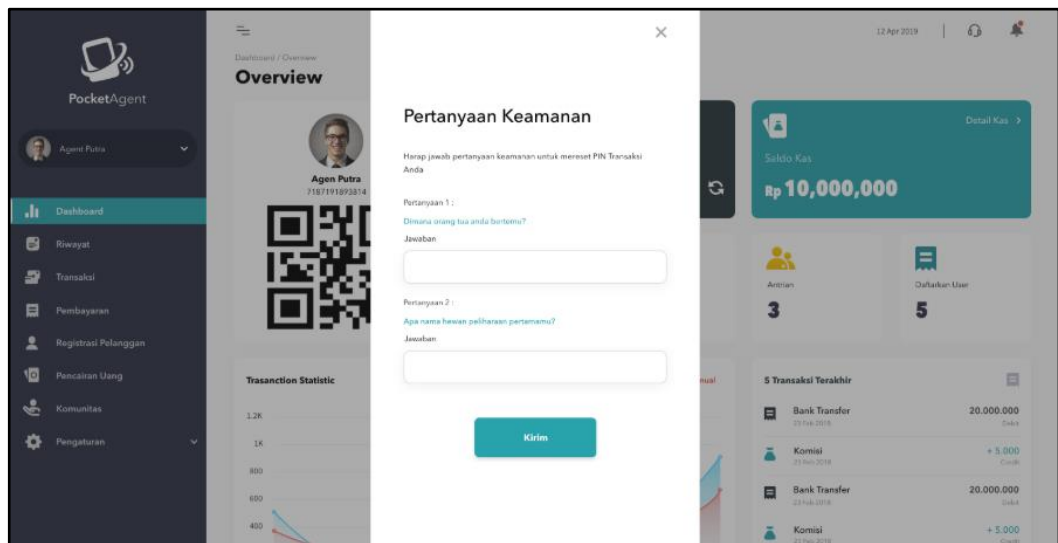
Ketika *user* menekan *link button* Lupa PIN pada aplikasi halaman Ubah PIN, maka akan muncul *modal* Lupa PIN. Dimana pada halaman ini, *user* diminta untuk memasukkan nama lengkap sesuai dengan nama lengkap yang pernah dimasukkan pada waktu mengaktifkan akun. Setelah itu, *user* dapat menekan *button* Kirim. Maka, secara otomatis aplikasi akan melakukan validasi nama lengkap sesuai dengan data sebelumnya. Gambar 3.101 menggambarkan halaman *modal input username* pada Forgot PIN.

Setelah selesai memasukkan nama lengkap dengan benar, maka *user* akan dikirimkan kode OTP pada nomor telepon yang didaftarkan pada saat *activation account*. Kemudian *user* diminta untuk memasukkan kode OTP tersebut ke kolom yang tersedia pada *modal verification*. Lalu *user* dapat menekan *button* Kirim. Secara otomatis, aplikasi akan melakukan validasi pada Kode OTP yang dimasukkan oleh *user*. Gambar 3.102 menggambarkan tampilan *modal* untuk

melakukan verifikasi kode OTP yang sudah dikirimkan ke nomor ponsel *user* yang terdaftar.



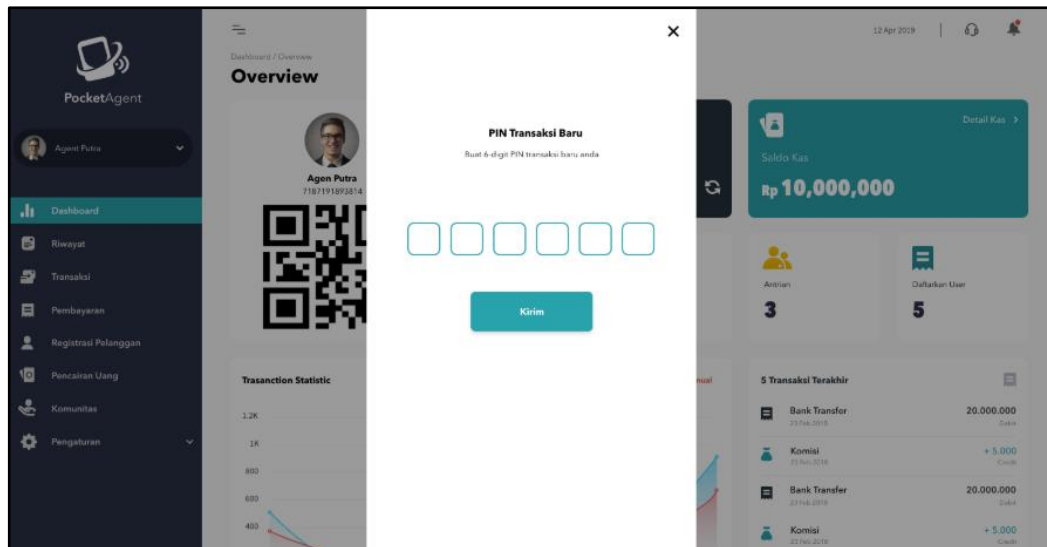
Gambar 3.102 Halaman Forgot PIN (Lanjutan)



Gambar 3.103 Halaman Forgot PIN (Lanjutan)

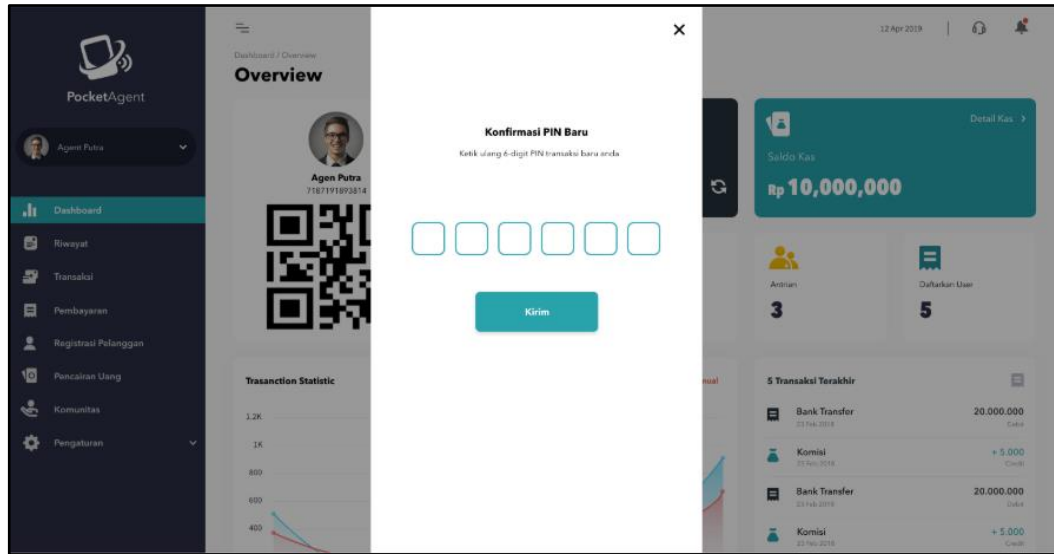
Setelah melakukan input kode OTP pada *modal verification*, maka *user* akan diberikan pertanyaan berupa *security question* yang pernah dipilih dan dijawab ketika melakukan *activation account*. *User* diminta untuk menjawab pertanyaan tersebut sesuai dengan jawaban yang diberikan ketika melakukan *activation account*. Lalu *user* dapat menekan *button* Kirim untuk melanjutkan proses *reset*

PIN transaksi. Gambar 3.103 menggambarkan *modal* untuk menjawab *security question* yang pernah dijawab oleh *user* ketika melakukan *activation account*.

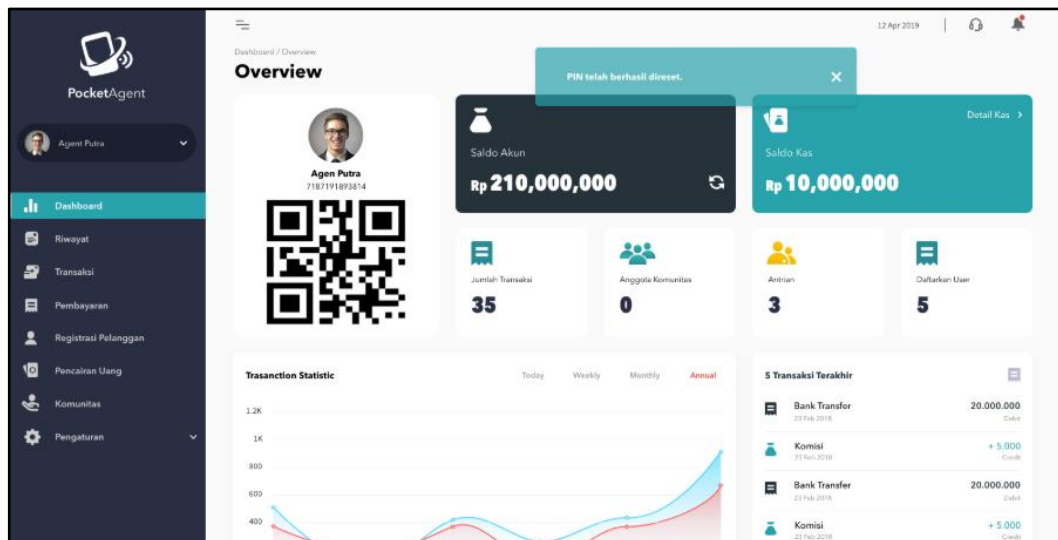


Gambar 3.104 Halaman *Forgot PIN* (Lanjutan)

Setelah semua proses validasi dilakukan dengan benar, maka *user* dapat memasukkan PIN transaksi yang baru dengan memasukkan 6 digit angka sesuai dengan keinginan *user* dan menekan *button* Kirim. Gambar 3.104 menggambarkan *modal* untuk membuat PIN transaksi baru. Setelah itu, *user* diminta untuk melakukan konfirmasi PIN yang baru dibuat untuk memastikan PIN transaksi baru yang diketik sebelumnya benar. Setelah itu, *user* dapat menekan *button* Kirim kembali. Gambar 3.105 menggambarkan *modal* Konfirmasi PIN baru dimana *user* wajib memasukkan kembali PIN transaksi yang baru dibuat sebagai tanda konfirmasi terhadap PIN baru.



Gambar 3.105 Halaman Forgot PIN (Lanjutan)



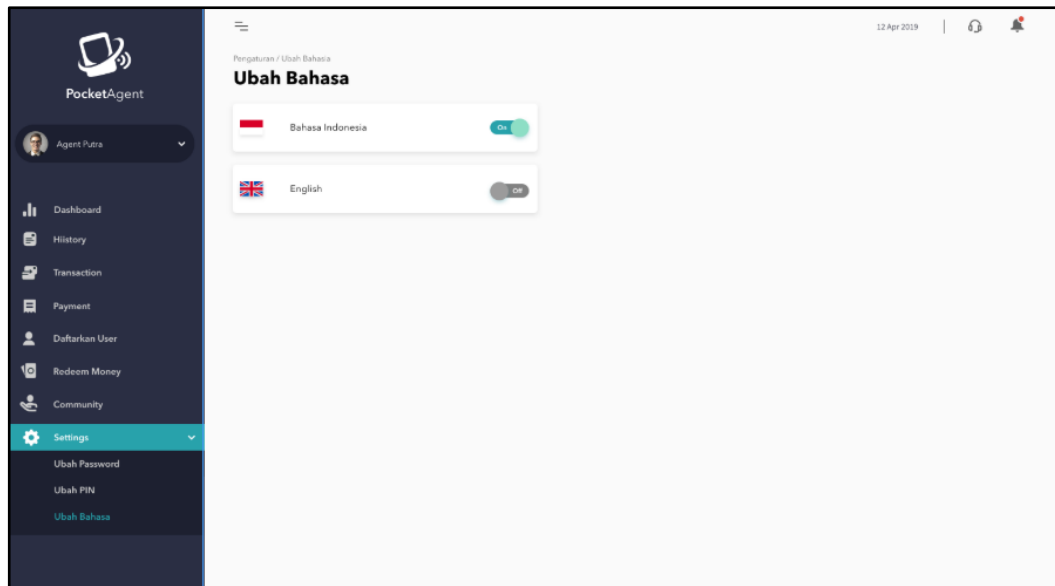
Gambar 3.106 Halaman Forgot PIN (Lanjutan)

Setelah *user* berhasil mengganti PIN, maka aplikasi akan mengeluarkan notifikasi yang menandakan PIN berhasil direset dan *user* dapat langsung menggunakan PIN transaksi yang baru untuk melakukan transaksi selanjutnya. Gambar 3.106 menggambarkan ketika *forgot* PIN berhasil dilakukan.

11.4. Halaman *Mock Up Change Language*

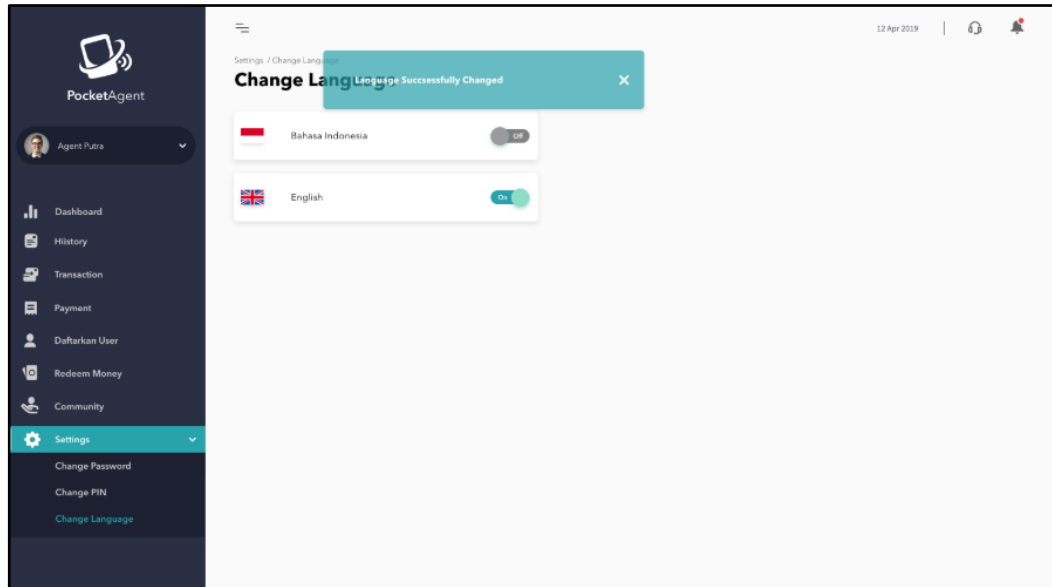
Gambar 3.107 sampai dengan Gambar 3.109 menunjukkan rancangan halaman *Change Language* yang dapat digunakan *user* untuk merubah bahasa

sesuai keinginan. Dalam fitur ini terdapat dua jenis bahasa yaitu Bahasa Indonesia dan Bahasa Inggris

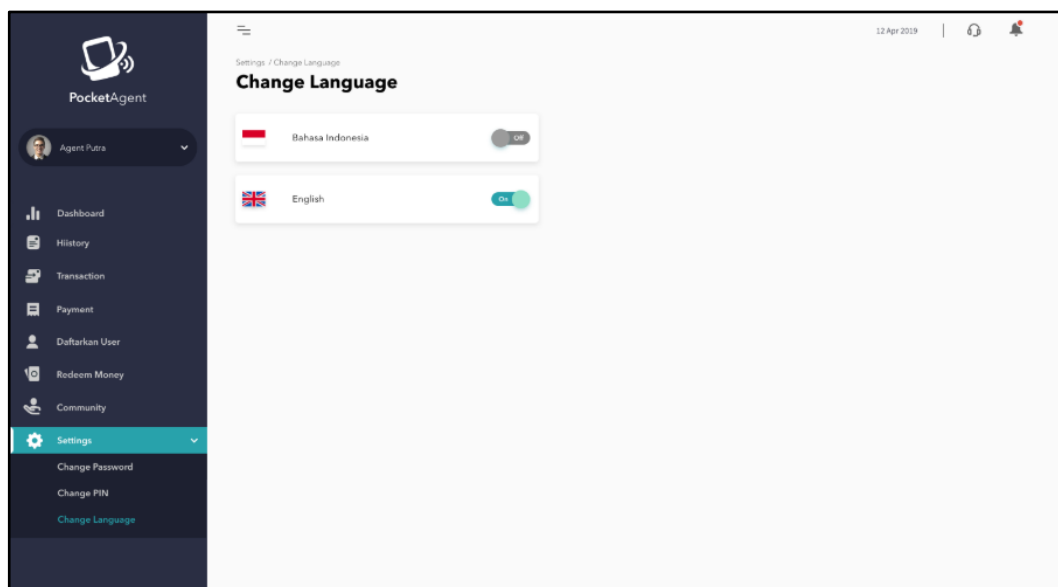


Gambar 3.107 Halaman Mock Up Change Language

Gambar 3.107 menggambarkan halaman *change language*. Dimana dalam aplikasi Pocketbank Agent terdapat dua jenis bahasa yaitu Bahasa Indonesia dan *English* untuk Bahasa Inggris yang dapat *user* pilih dengan menekan *switch on* dan *switch off* pada aplikasi. Ketika *user* melakukan *switch off* pada Bahasa Indonesia, maka secara otomatis bahasa berubah menjadi *English* dan begitu juga sebaliknya. Aplikasi Pocketbank Agent memiliki Bahasa Indonesia sebagai bahasa *default* aplikasi. Ketika *user* mengganti bahasa, maka aplikasi akan menampilkan notifikasi sesuai dengan bahasa yang dipilih yang menandakan bahwa bahasa sudah diganti ke bahasa yang sesuai dengan pilihan *user*. Gambar 3.108 menggambarkan notifikasi menggambarkan tampilan *notification* ketika bahasa diubah.



Gambar 3.108 Halaman Mock Up Change Language (Lanjutan)

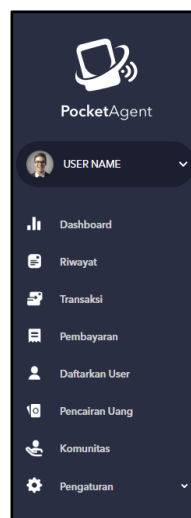


Gambar 3.109 Halaman Mock Up Change Language (Lanjutan)

Gambar 3.109 menggambarkan halaman *change language* yang sudah diubah bahasa. Dimana bahasa yang diubah adalah bahasa keseluruhan aplikasi dari semua fitur yang ada pada aplikasi termasuk halaman *login* pada aplikasi Pocketbank Agent. Bahasa yang digunakan harus disimpan pada *local storage* jika agar bahasa tidak berubah ketika melakukan *reload* pada halaman *website*.

3.3.2. Implementasi

Aplikasi Pocketbank Agent berbasis website dengan fitur yang telah selesai dibuat dan implementasikan. Namun, aplikasi tersebut masih terus dikembangkan dengan menambah fitur-fitur pada Aplikasi Pocketbank Agent oleh PT Anabatic Technologies Tbk sehingga *project* yang sudah selesai dibuat masih dalam keadaan *localhost*. Implementasi dari aplikasi tersebut akan dijelaskan dengan menggunakan gambar-gambar *screenshot*.



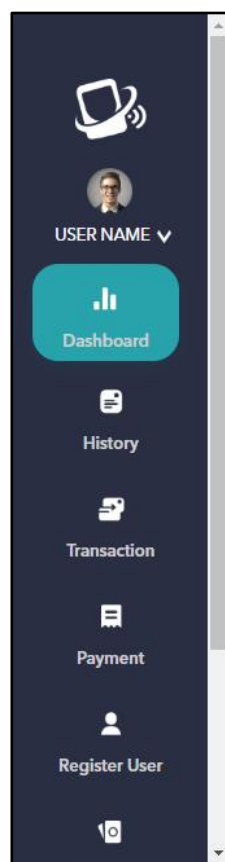
Gambar 3.110 Tampilan Menu Pocketbank Agent

Gambar 3.110 menunjukkan Menu yang tersedia pada Aplikasi Pocketbank Agent. Menu akan selalu berada pada sebelah kiri di setiap halaman dalam aplikasi ini. Dalam menu *side-bar* selalu terdapat logo Aplikasi Pocketbank Agent sebagai logo aplikasi. Menu dapat di-*minimize* sehingga hanya terdapat tampilan gambar *icon* sesuai dengan fiturnya. Gambar 3.111 menunjukkan tombol *icon* untuk *minimize sidebar* yang terdapat di setiap fitur pada Aplikasi Pocketbank Agent.



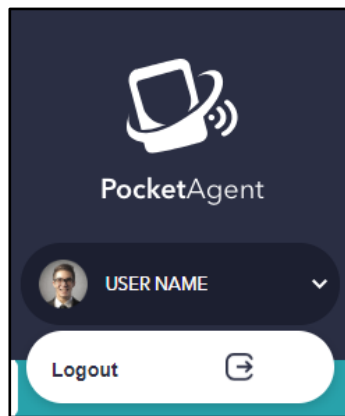
Gambar 3.111 Tampilan *Button Icon Minimize Sidebar*

Bila *user* menekan tombol pada Gambar 3.111 maka tampilan menu *sidebar* akan diperkecil, logo Aplikasi Pocketbank berubah menjadi tanpa judul aplikasi dan hanya terdapat *icon* setiap fitur beserta nama fitur dan terdapat *scrollbar* pada menu tampilan menu yang dan terdapat *border* berwarna biru sebagai penanda *user* sedang berada pada fitur tertentu. Gambar 3.112 menunjukkan tampilan menu *sidebar* yang diperkecil.

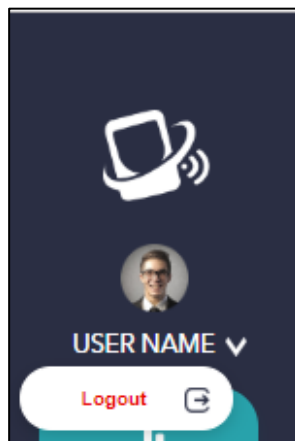


Gambar 3.112 Tampilan Menu *Sidebar* yang diperkecil

Dalam tampilan menu *sidebar* terdapat *profile user* beserta fotonya dan *username*. Pada *username* terdapat *button dropdown* yang berisi fungsi *logout*. Gambar 3.113 menunjukkan tampilan *dropdown* pada *username* yang berisi fungsi *logout* ketika menu tidak di *minimize* dan Gambar 3.114 menunjukkan tampilan menu *dropdown* pada *username* yang berisi fungsi *logout* ketika menu di-*minimize*.

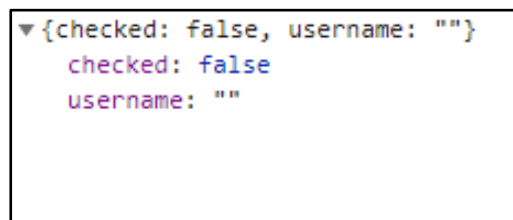


Gambar 3.113 Tampilan *Dropdown* pada Username

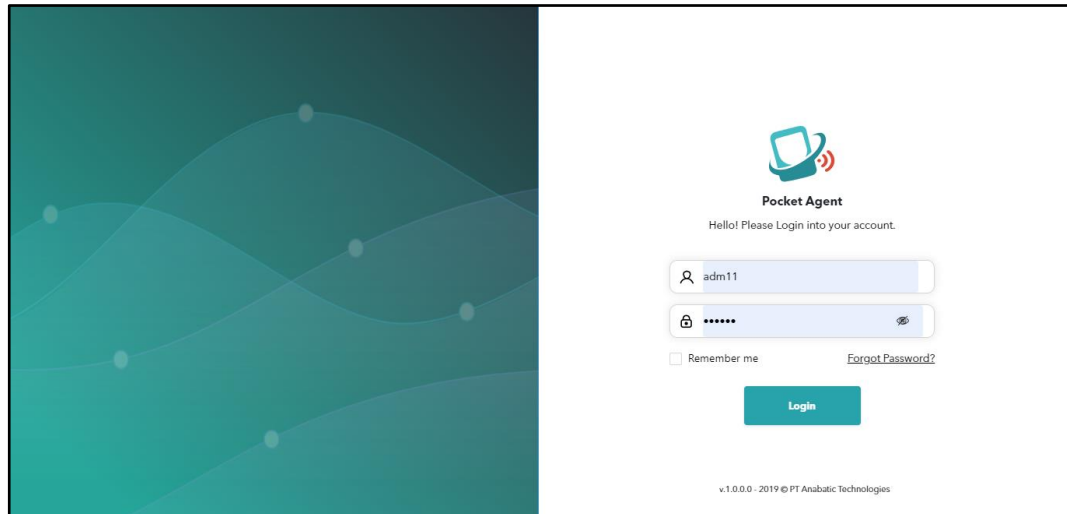


Gambar 3.114 Tampilan Minimize *Dropdown* pada Username

Ketika *user* menekan tombol *logout* maka akan dikembalikan kedalam halaman *login*. Gambar 3.115 menunjukkan tampilan ketika *user* melakukan *logout*, dimana dalam fungsi *logout* semua data *user* telah dihapus dari *local storage*. Gambar 3.116 menunjukkan tampilan setelah *user* melakukan *logout*.

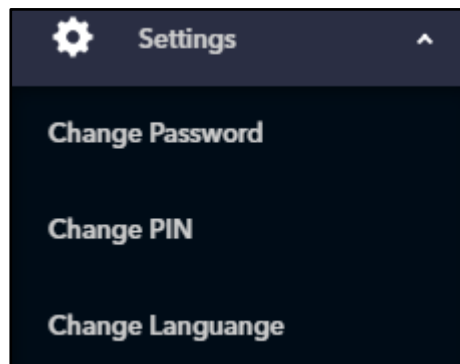


Gambar 3.115 Tampilan *Local Storage* ketika *logout*



Gambar 3.116 Tampilan Setelah Melakukan *Logout*

Dalam Aplikasi Pocketbank Agent terdapat menu *setting* yang terdiri dari *Change Language*, *Change PIN*, dan *Change Password* yang memiliki fungsinya masing-masing. Gambar 3.117 menunjukkan tampilan Menu *Setting* pada Aplikasi Pocketbank Agent.



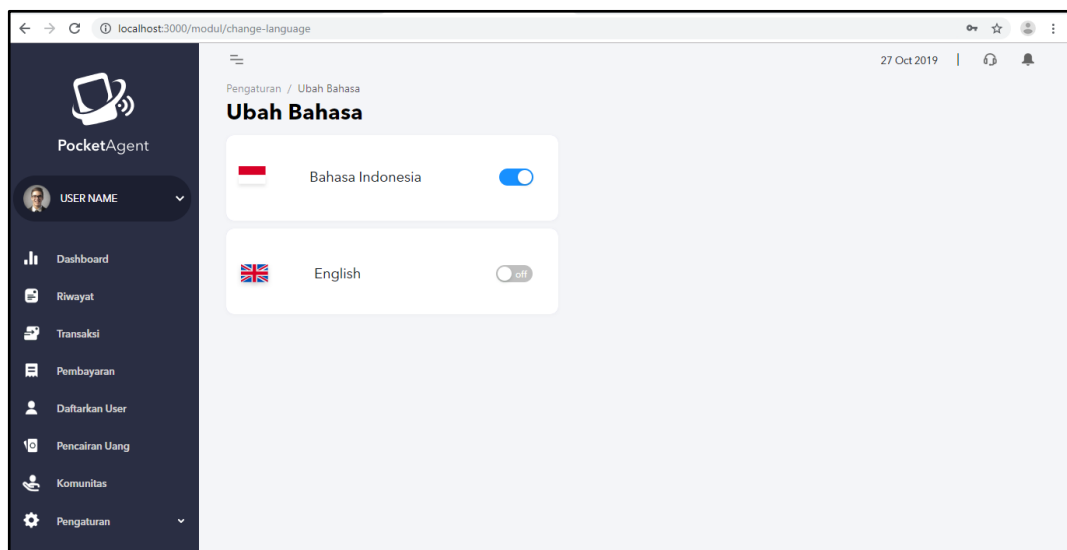
Gambar 3.117 Tampilan Menu *Setting* Pocketbank Agent

Dalam fitur *Change Language* terdiri dari tampilan *breadcrumb* yang terdiri dari *button menu sidebar*, *current date*, *header*, *subheader*, *contact*, dan *notification*. Gambar 3.118 menunjukkan tampilan *breadcrumb* fitur *Change Language*.



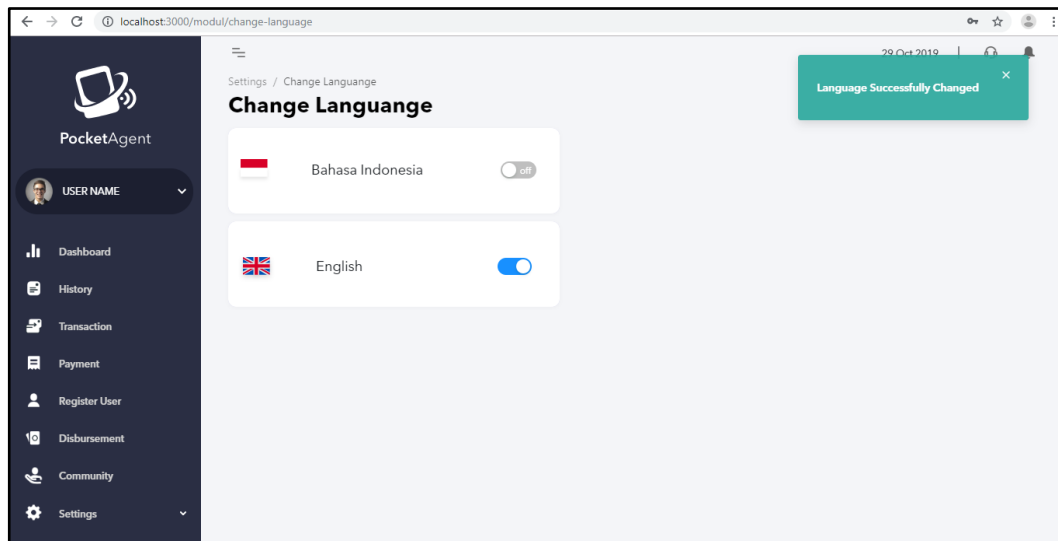
Gambar 3.118 Tampilan *breadcrumb* Change Language

Pada fitur *Change Language* terbagi menjadi dua *grid layout* Bahasa Indonesia dan *grid layout English* yang masing masing terdapat logo bendera dan *switch button on-off*. Gambar 3.119 menunjukkan tampilan fitur *Change Language* pada Aplikasi Pocketbank Agent.



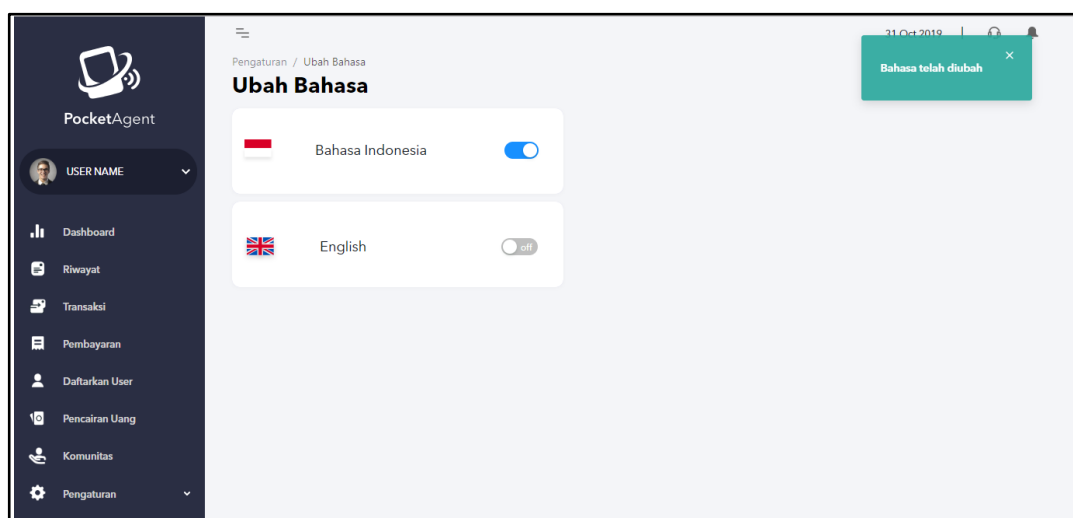
Gambar 3.119 Tampilan Fitur Change Language

Ketika *button toggle switch* pada posisi *on* pada Bahasa Indonesia ditekan, maka *button toggle switch* pada Bahasa Indonesia langsung berwarna abu-abu (*off*) mengeluarkan *alert toast* yang sesuai dengan bahasanya dan bahasa langsung berubah menjadi *English*. Begitu juga sebaliknya bila *button toggle switch English* ditekan maka terjadi hal yang sama pada Bahasa Indonesia. Gambar 3.120 menunjukkan fitur *Change Language* jika Bahasa Indonesia diganti menjadi *English*.



Gambar 3.120 Tampilan Change Language *English* Diubah

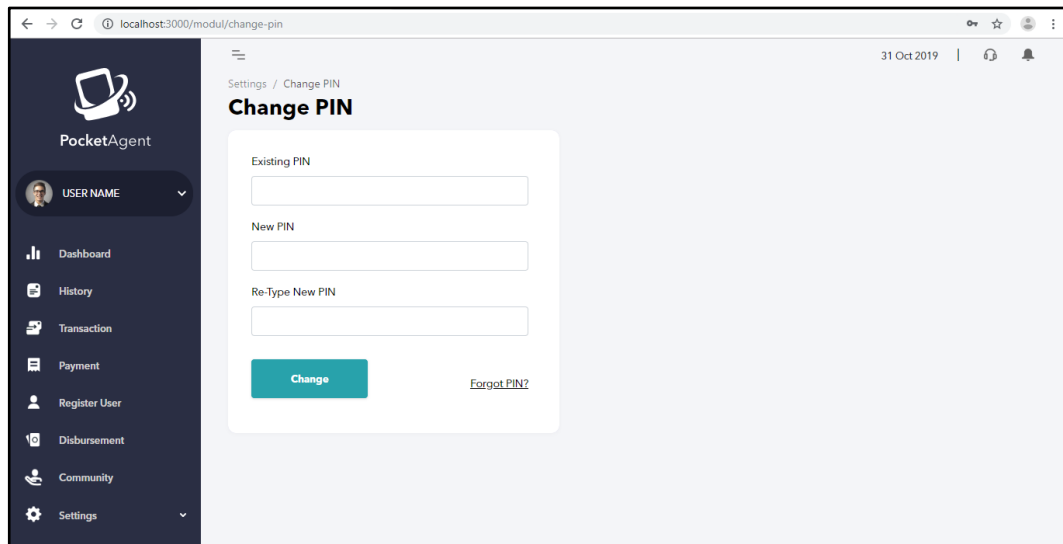
Fitur ubah bahasa yang dibuat sudah sesuai dengan standar bahasa pada umumnya, sehingga tidak menggunakan *library* bahasa secara khusus untuk mengimplementasi fitur ubah bahasa. Ketika bahasa diubah, maka aplikasi akan mengeluarkan notifikasi yang menyatakan bahwa bahasa telah berhasil diubah sesuai dengan bahasa yang dipilih oleh *user*. Gambar 3.121 menunjukkan tampilan ketika Fitur Bahasa Indonesia kembali diaktifkan.



Gambar 3.121 Tampilan Ketika Bahasa Indonesia Diaktifkan

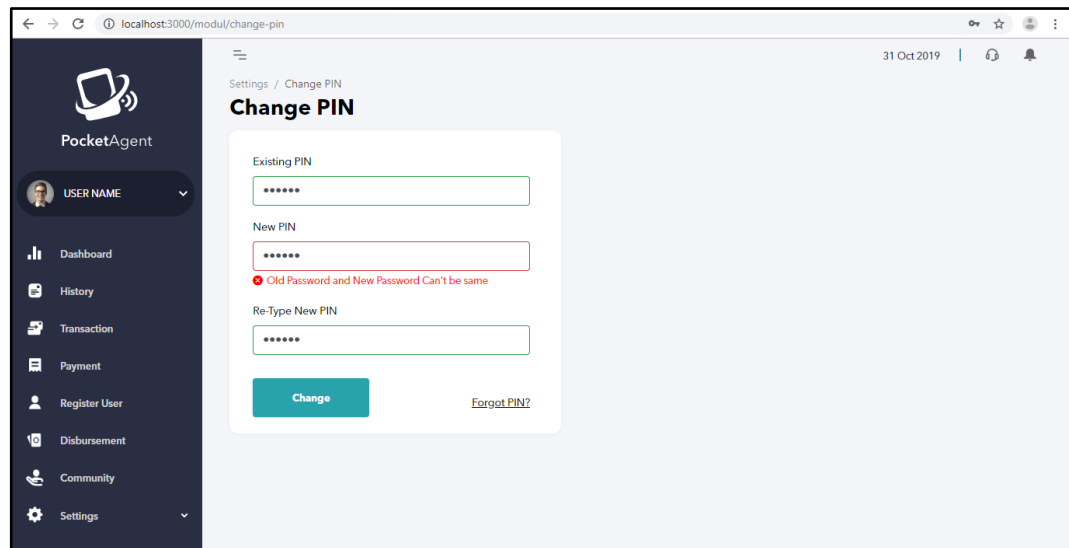
Selain Fitur *Change Language* pada Menu *Settings* juga terdapat fitur *Change PIN* dan *Change Password*. Fitur *Change PIN* dapat digunakan ketika *user*

ingin mengganti PIN transaksi pada aplikasi dengan memasukkan *Existing* PIN lalu menggantinya dengan PIN yang baru. Fitur *Change* PIN terdiri dari sebuah *grid* yang berisi *form*. Gambar 3.122 menunjukkan tampilan fitur *Change* PIN pada aplikasi Pocketbank Agent.

The image is a screenshot of a web browser displaying the 'Change PIN' feature of the PocketAgent application. The browser's address bar shows 'localhost:3000/modul/change-pin'. The application has a dark blue sidebar on the left with the 'PocketAgent' logo and a list of menu items: 'Dashboard', 'History', 'Transaction', 'Payment', 'Register User', 'Disbursement', 'Community', and 'Settings'. The main content area is light blue and titled 'Settings / Change PIN' and 'Change PIN'. It contains a white form with three input fields: 'Existing PIN', 'New PIN', and 'Re-Type New PIN'. Below the 'Re-Type New PIN' field is a teal 'Change' button and a link that says 'Forgot PIN?'. The top right of the application shows the date '31 Oct 2019' and some notification icons.

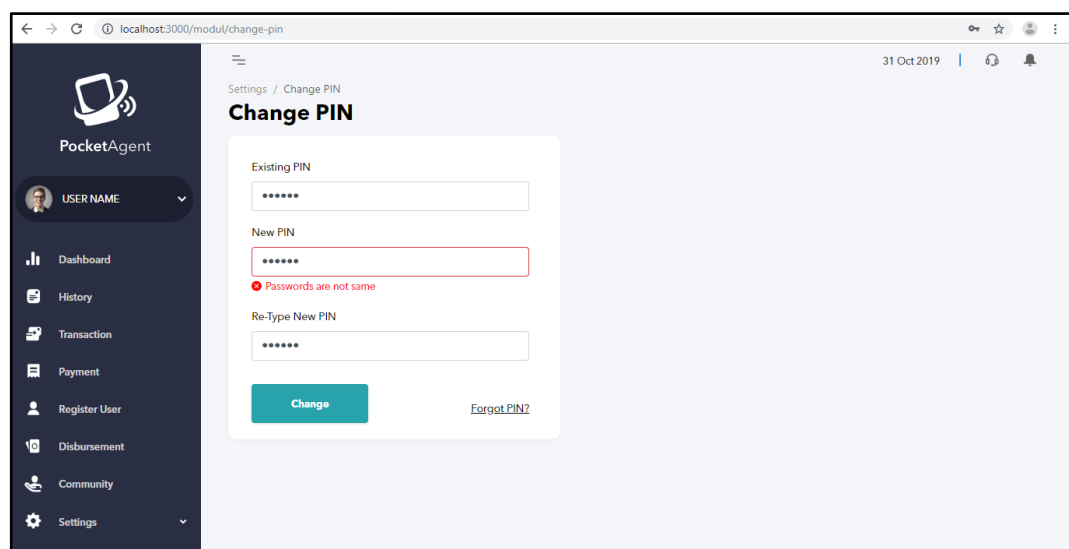
Gambar 3.122 Tampilan Fitur Change PIN

Pada fitur *Change* PIN sudah terdapat beberapa validasi khusus meningkatkan *security* atau keamanan pada aplikasi bank. Dimana pada Aplikasi Pocketbank Agent, *user* tidak dapat menggunakan PIN yang sama ketika ingin mengganti PIN yang baru dan PIN yang sudah pernah digunakan tidak boleh digunakan sementara (sebanyak satu kali) ketika ingin mengganti PIN transaksi berikutnya. Gambar 3.123 menunjukkan Tampilan Validasi *Change* PIN ketika PIN sekarang sama dengan PIN yang baru dibuat.



Gambar 3.123 Tampilan Validasi Change PIN

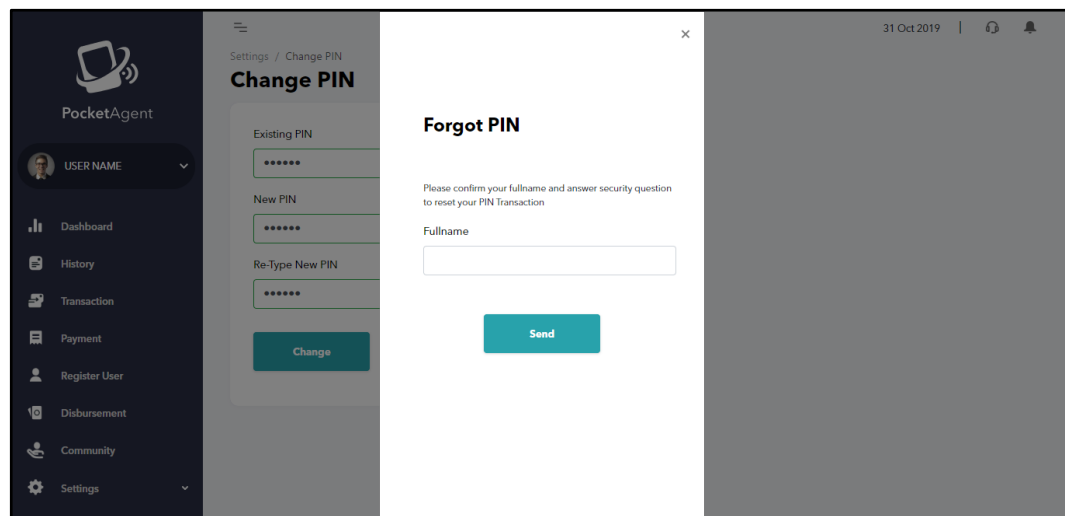
Selain itu *Change PIN* juga terdapat validasi khusus dimana, PIN baru yang diketik harus sama dengan PIN konfirmasi yang diketik. Bila tidak sama, maka Aplikasi akan mengeluarkan validasi *password* tidak sama terhadap Fitur *Change PIN*. Proses pengetikan sebanyak dua kali bertujuan untuk melakukan konfirmasi ulang terhadap PIN yang baru dibuat oleh *user*. Gambar 3.124 menunjukkan tampilan validasi *password* tidak sama pada Fitur *Change PIN*.



Gambar 3.124 Tampilan Validasi Change PIN (lanjutan)

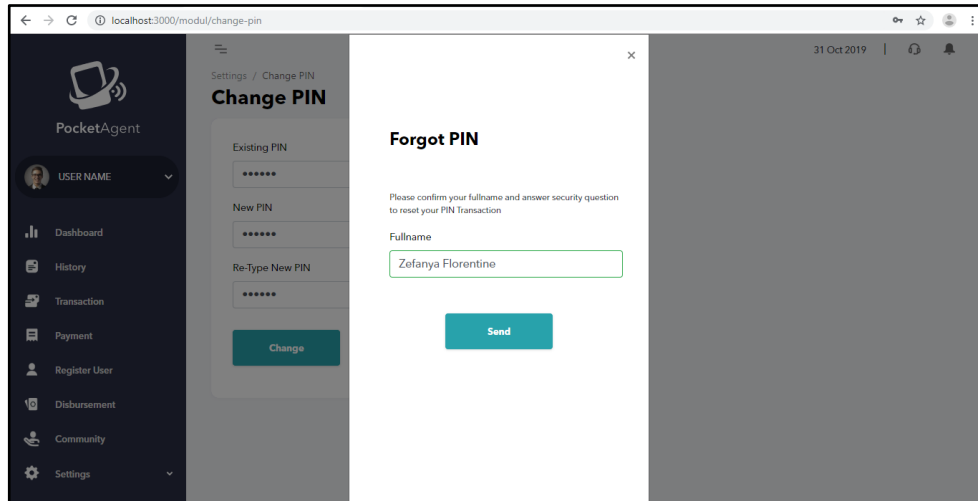
Pada Fitur *Change PIN* terdapat Fitur *Forgot PIN* dimana Fitur *Forgot PIN*

dapat digunakan ketika *user* lupa PIN transaksi yang digunakan. Tampilan Fitur *Forgot PIN* berupa *modal* yang dikeluarkan pada Fitur *Change PIN*. Fitur *Forgot PIN* dapat digunakan *user* untuk melakukan *reset* pada PIN Transaksi apabila *user* benar- benar lupa akan PIN Transaksi yang dibuat ketika melakukan *activation account*. Gambar 3.125 menunjukkan tampilan Fitur *Forgot PIN* pada Aplikasi Pocketbank Agent.



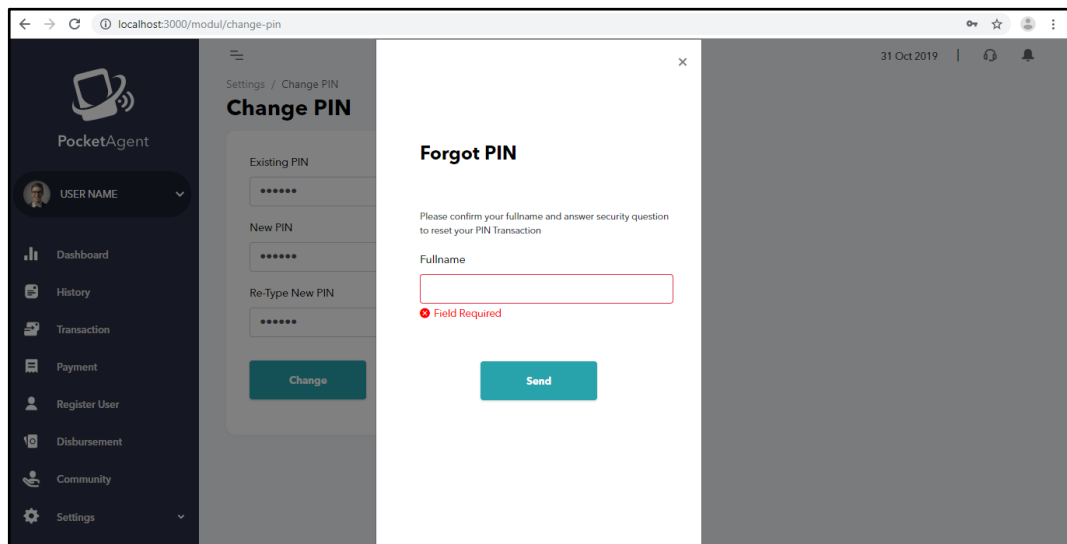
Gambar 3.125 Tampilan *Forgot PIN*

Dalam Fitur *Forgot PIN* pertama *user* akan diminta untuk memasukkan nama lengkap sesuai dengan yang terdaftar pada aplikasi pada waktu melakukan *activation account*, setelah itu menekan *button send* untuk melanjutkan proses *Forgot PIN*. Apabila *user* menekan *button close* pada sudut kanan atas *modal Forgot PIN* maka semua tindakan dibatalkan. Secara otomatis, aplikasi akan melakukan validasi akan nama lengkap yang dimasukkan oleh *user*. Gambar 3.126 menunjukkan tampilan *input* nama lengkap pada *Forgot PIN*.



Gambar 3.126 Tampilan *Input Fullname* Forgot PIN

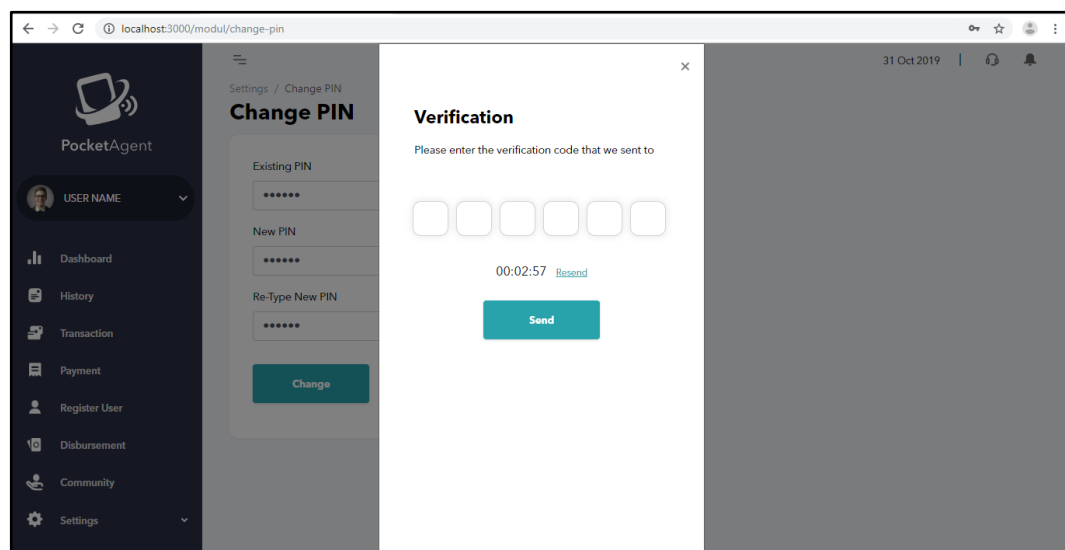
Pada Fitur *Forgot PIN* juga terdapat validasi dimana jika *user* menekan *button send* ketika belum mengisi *form fullname* maka *user* tidak dapat melanjutkan ke tahap selanjutnya untuk melakukan *reset PIN* transaksi. Validasi yang dilakukan hanya berupa *field required* dimana *user* wajib mengisi nama lengkap pada *modal username* pada *Forgot PIN*. Gambar 3.127 menunjukkan validasi yang ada pada *modal Forgot PIN*.



Gambar 3.127 Tampilan Validasi Fullname Forgot PIN

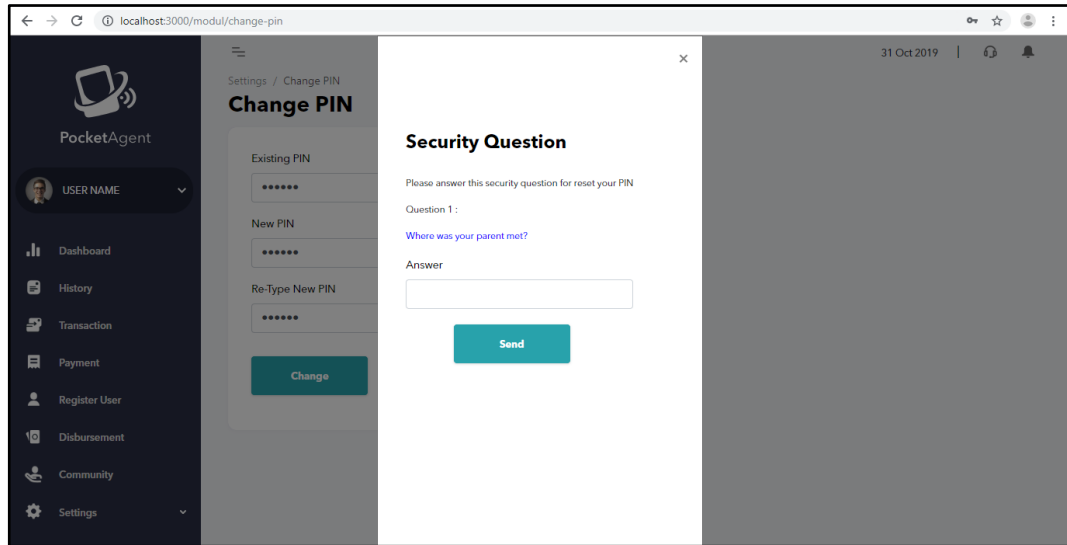
Ketika *user* sudah melakukan *input* nama lengkapnya dan menekan *button send*, maka akan diberikan *modal form verification* dan diminta untuk memasukan

kode OTP yang diberikan pada nomor ponsel yang didaftarkan pada aplikasi ketika melakukan *activation account*. Bila *user* tidak mendapatkan kode OTP yang dikirimkan ke nomor ponsel *user*, maka *user* dapat meminta kirim ulang terhadap kode OTP yang dikirim dengan menekan *link button resend*, maka secara otomatis, kode OTP akan terkirim kembali. Gambar 3.128 menunjukkan tampilan *verification* kode OTP pada Fitur *Change PIN*.



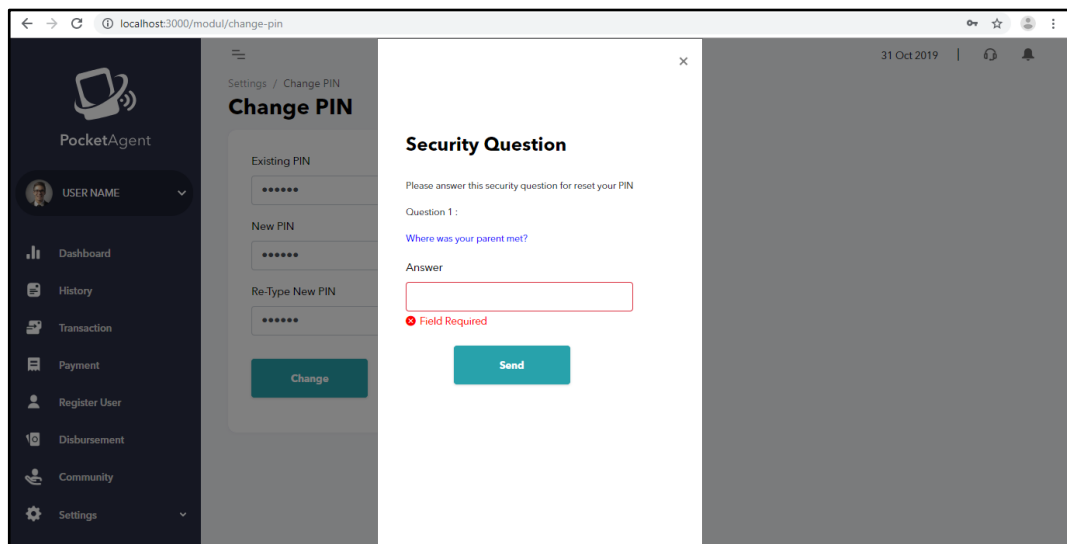
Gambar 3. 128 Tampilan Verification pada Fitur Change PIN

Setelah selesai memasukan kode OTP dan menekan *button* Send, secara otomatis aplikasi akan mengirimkan kode OTP yang diinput oleh *user* dengan kode OTP yang dikirim pada nomor ponsel *user*. Setelah kode OTP *valid*, maka *user* diminta untuk memasukan validasi berikutnya berupa *security question* yang pernah dipilih dan dijawab ketika melakukan *activation account*. Gambar 3.129 menunjukkan tampilan *Security Question* pada fitur *Forgot PIN*.



Gambar 3.129 Tampilan Security Question Fitur Forgot PIN

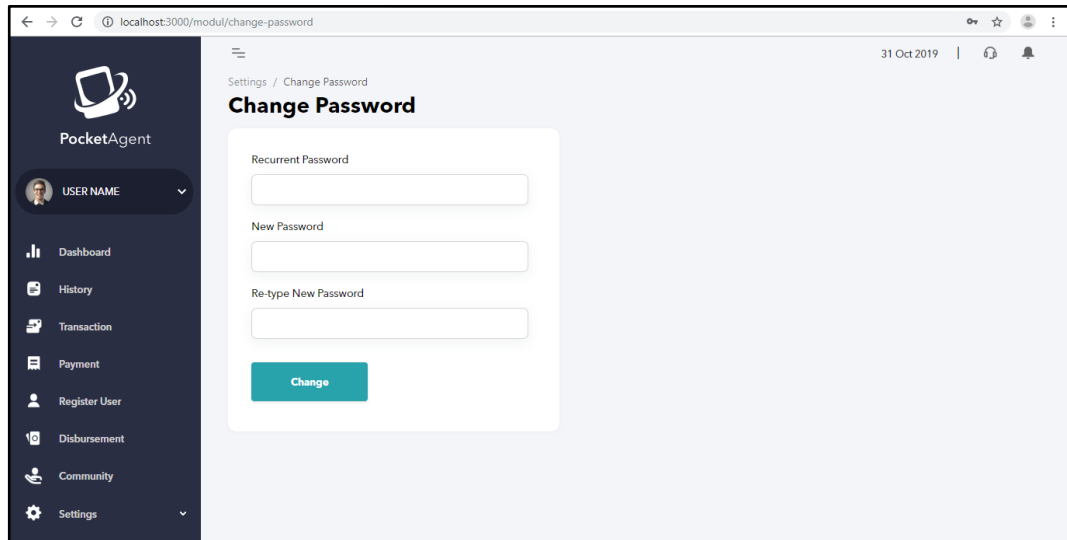
Pada *Security Question* , sudah terdapat validasi *field required* jika *form answer* kosong setelah menekan *button send*. Dimana *user* wajib menjawab *security question* yang ada pada *modal Security Question* untuk melanjutkan proses *reset PIN* transaksi. Gambar 3.130 menunjukkan tampilan validasi pada *Security Question* apabila *form* kosong.



Gambar 3.130 Tampilan Validasi Security Question

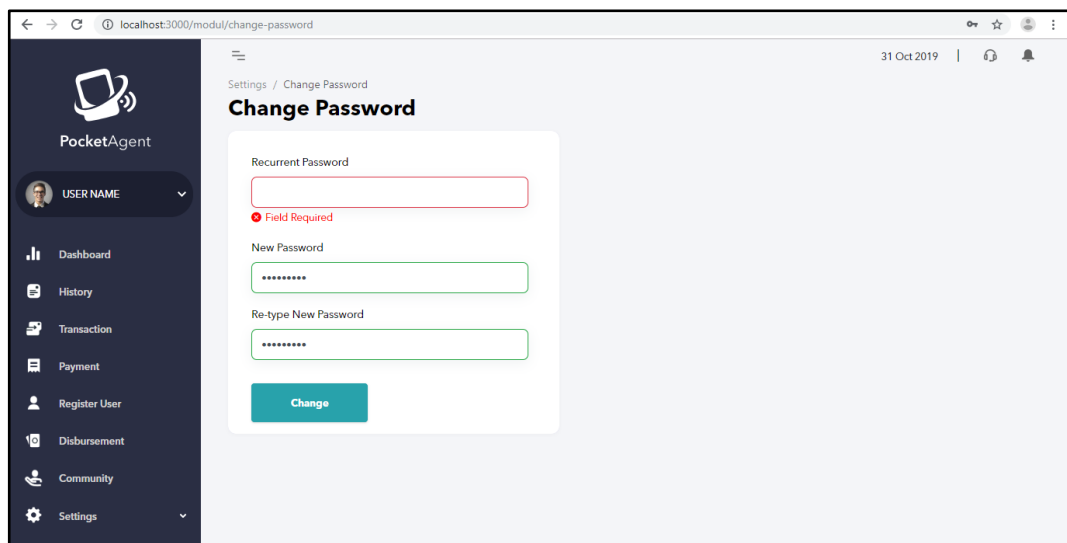
Fitur *Change Password* juga terdapat pada Menu *Settings*, dimana *Change Password* digunakan ketika *user* ingin mengganti *password* pada akunnya. Fitur

Change Password terdiri dari sebuah *grid* yang berisi *form input* yang sama seperti pada fitur *Change PIN*. Gambar 3.131 menunjukkan tampilan Fitur *Change Password* pada Menu *Settings*.

The screenshot shows a web browser at localhost:3000/modul/change-password. The interface has a dark sidebar on the left with the 'PocketAgent' logo and a menu including Dashboard, History, Transaction, Payment, Register User, Disbursement, Community, and Settings. The main content area is titled 'Settings / Change Password' and 'Change Password'. It contains three input fields: 'Recurrent Password', 'New Password', and 'Re-type New Password'. A green 'Change' button is positioned below the input fields. The browser's address bar and top navigation bar are visible.

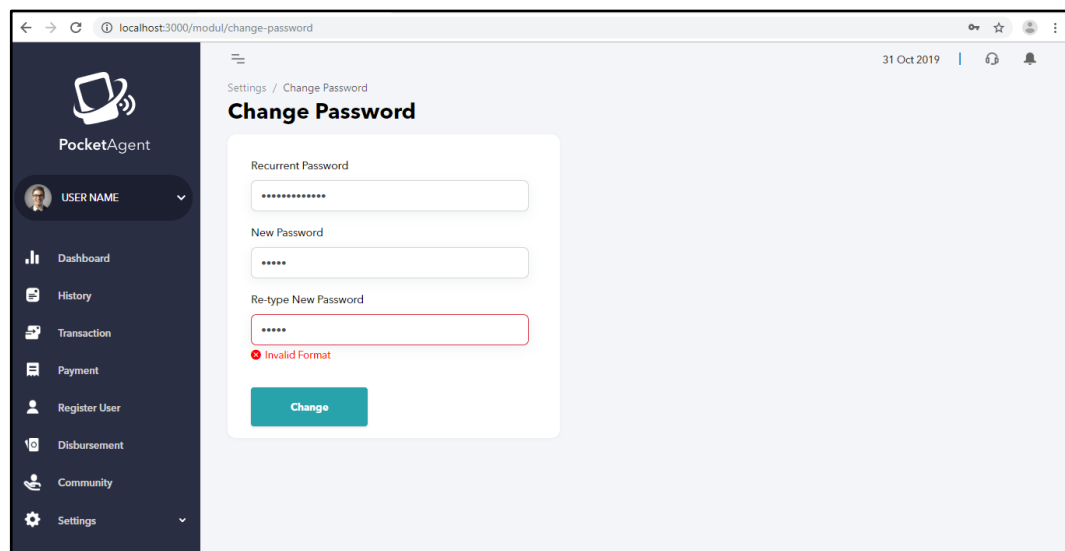
Gambar 3.131 Tampilan Fitur Change Password

Pada Fitur *Change Password* juga dilengkapi dengan validasi *field required* pada setiap *form input* yang ada pada Fitur *Change Password*. Dimana *form* yang diinput tidak boleh kosong baik pada salah satu maupun seluruh *form*. Gambar 3.132 menunjukkan tampilan validasi pada Fitur *Change Password*.

This screenshot shows the same 'Change Password' form as in Gambar 3.131, but with validation errors. The 'Recurrent Password' field is highlighted with a red border and has a red error message 'Field Required' below it. The 'New Password' and 'Re-type New Password' fields have green borders and are filled with asterisks. The 'Change' button remains green. The sidebar and browser interface are consistent with the previous image.

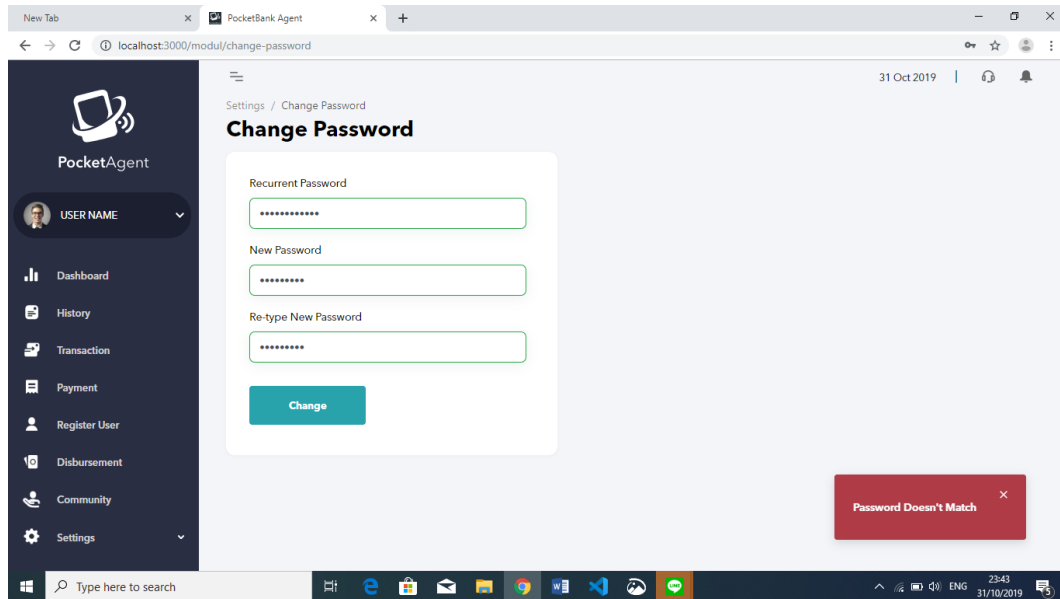
Gambar 3.132 Tampilan Validasi Forgot Password

Password baru yang diinput harus terdiri dari lebih dari 6 karakter dimana harus memiliki satu huruf kapital dan satu angka. Jika *password* yang dimasukkan tidak memenuhi kriteria, maka akan mengeluarkan validasi *format invalid*. Gambar 3.133 menunjukkan tampilan *invalid format* pada Fitur *Change Password*.



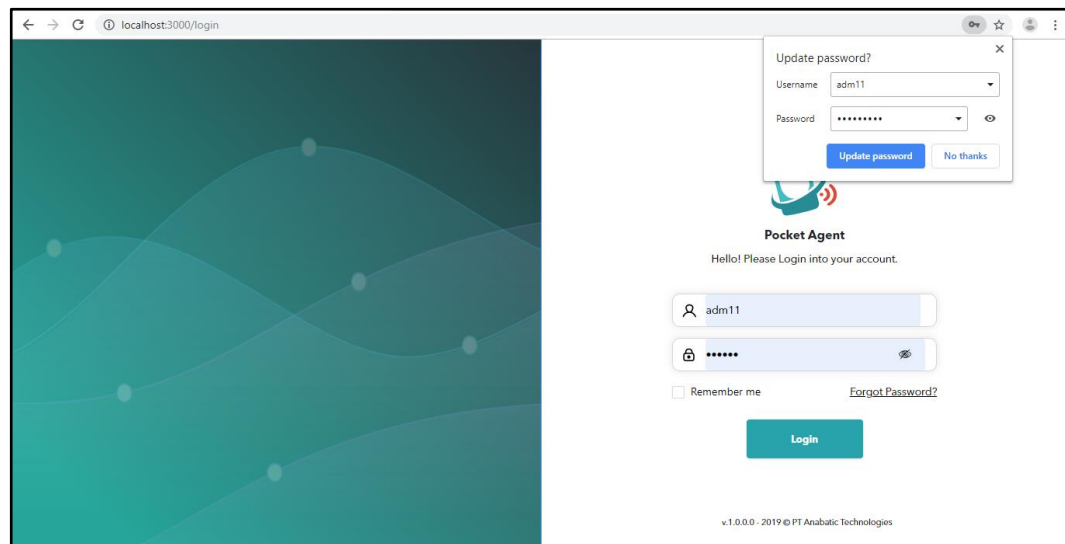
Gambar 3.133 Tampilan validasi Forgot Password (Lanjutan)

Selain itu, pada Fitur *Change Password*, aplikasi juga melakukan validasi untuk melakukan validasi terhadap password lama yang diinput oleh *user*. Jika *password* lama yang diinput *user* salah, maka aplikasi akan mengeluarkan *alert* berupa *notification* yang menyatakan bahwa *password* lama yang diinput oleh *user* salah. Gambar 3.134 menunjukkan tampilan *alert notification* merah yang menyatakan *password* yang dimasukkan oleh *user* salah.



Gambar 3.134 Tampilan Validasi Forgot Password (Lanjutan)

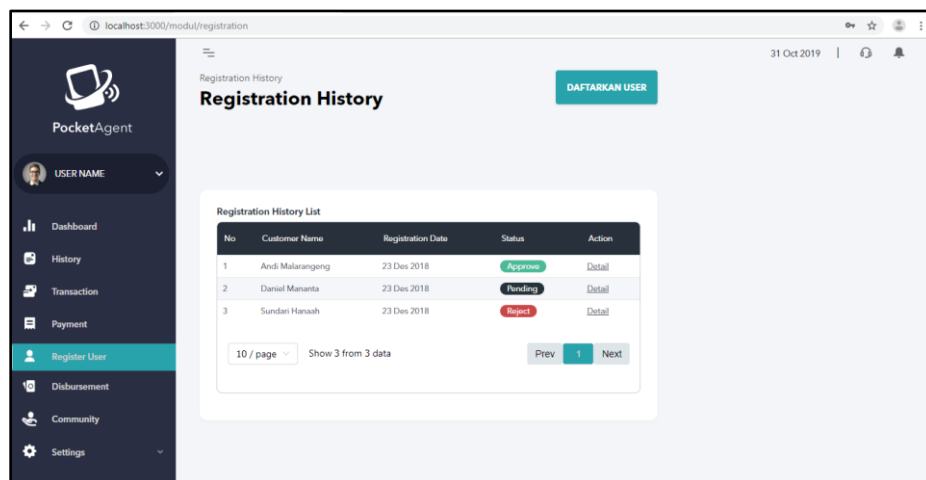
Bila *password* yang diinput sudah memenuhi kriteria, maka fitur akan menampilkan *alert* hijau disertai langsung ke halaman *login* dan melakukan *login* dengan menggunakan *password* yang baru. Gambar 3.135 menunjukkan Tampilan ketika *Change Password* yang dilakukan berhasil.



Gambar 3. 135 Tampilan Change Password berhasil

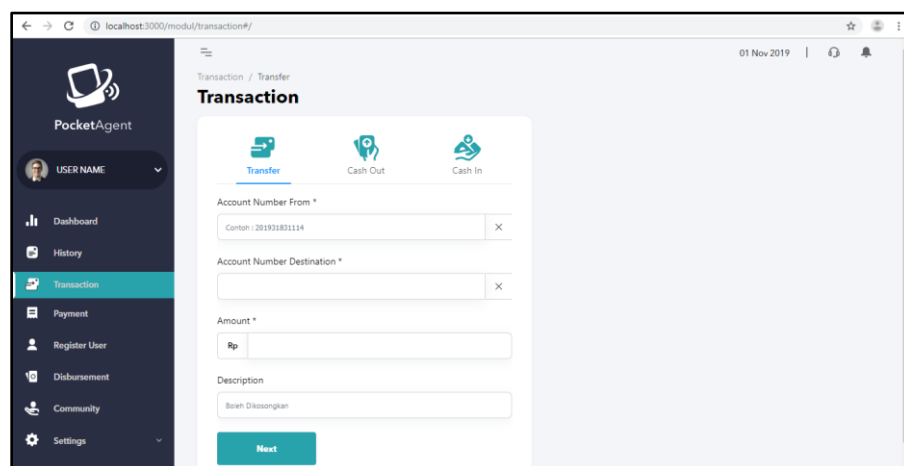
Pada Fitur *Registration* terdapat *registration history* dimana *user* dapat

melihat seluruh *list registration* yang pernah dilakukan beserta statusnya. Pada fitur ini, *user* dapat mendaftarkan *user* lain dengan cara manual maupun dengan melakukan *scan e-ktip*. Gambar 3.136 menunjukkan tampilan Fitur *Registration History*.



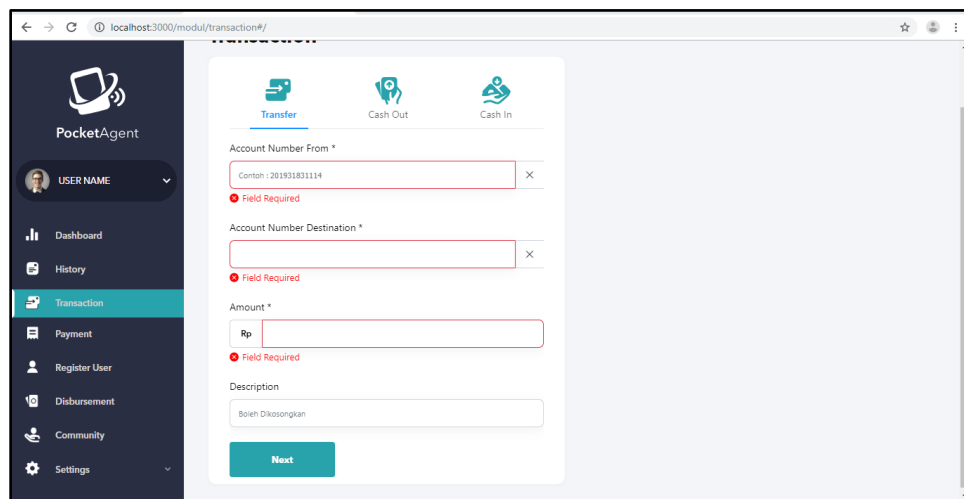
Gambar 3.136 Tampilan Registration History

Pada Fitur *Transaction* terdapat Fitur *transfer*, *cash-in*, dan *cash-out*. Dimana Fitur tersebut memiliki fungsi masing-masing. Pada Fitur *transfer*, *user* dapat menggunakan untuk mentransfer ke sesama pemilik *account* dengan nominal yang diinginkan. Gambar 3.137 menunjukkan tampilan Fitur *Transfer* pada *Transaction*.



Gambar 3.137 Tampilan Fitur Transfer

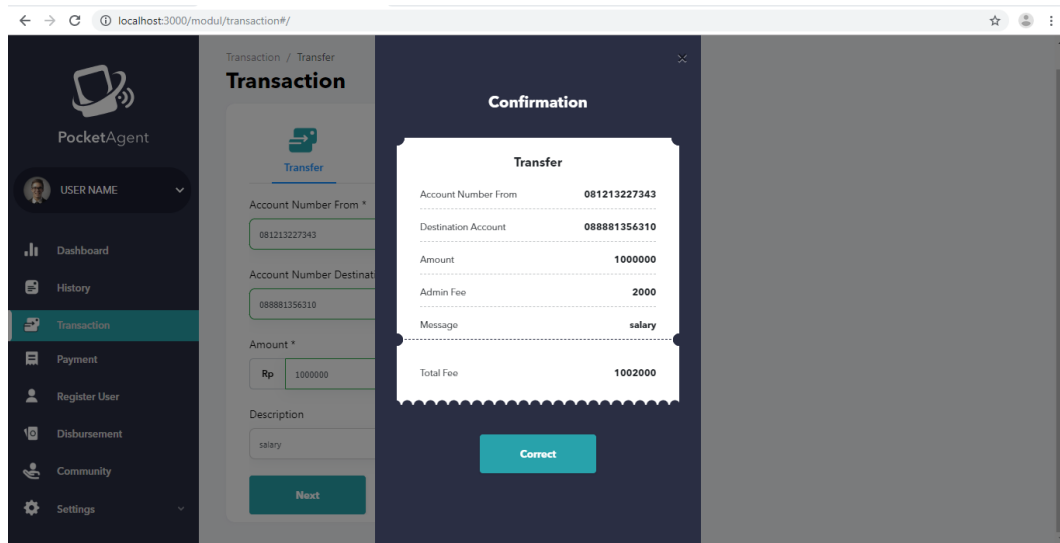
Pada Fitur *Transfer* juga sudah dilengkapi dengan validasi misalnya seperti *invalid format* dan *field required*. Dimana pada bagian *transfer* terdapat *form input* yang berisi Nomor Rekening Asal, Nomor Rekening Tujuan dan Jumlah uang yang ingin ditransfer yang harus diinput *user* berupa angka, sedangkan pada *form input* Deskripsi bisa berupa angka maupun *string character*. Semua *form input* pada bagian transfer wajib diisi oleh *user* kecuali pada *form input* deskripsi dapat diisi oleh *user* secara *optional*. Gambar 3.138 menunjukkan tampilan validasi Fitur *Transfer*.

The image is a screenshot of a web application interface for 'PocketAgent'. On the left is a dark sidebar with a menu containing 'Dashboard', 'History', 'Transaction' (highlighted), 'Payment', 'Register User', 'Disbursement', 'Community', and 'Settings'. The main content area has three tabs: 'Transfer' (active), 'Cash Out', and 'Cash In'. The 'Transfer' form contains the following fields: 'Account Number From *' with a red border and a red error message 'Field Required' below it; 'Account Number Destination *' with a red border and a red error message 'Field Required' below it; 'Amount *' with a red border and a red error message 'Field Required' below it; and 'Description' with a placeholder text 'Boleh Dikosongkan'. A teal 'Next' button is at the bottom of the form. The browser's address bar shows 'localhost:3000/module/transaction#/'.

Gambar 3.138 Tampilan validasi Fitur Transfer

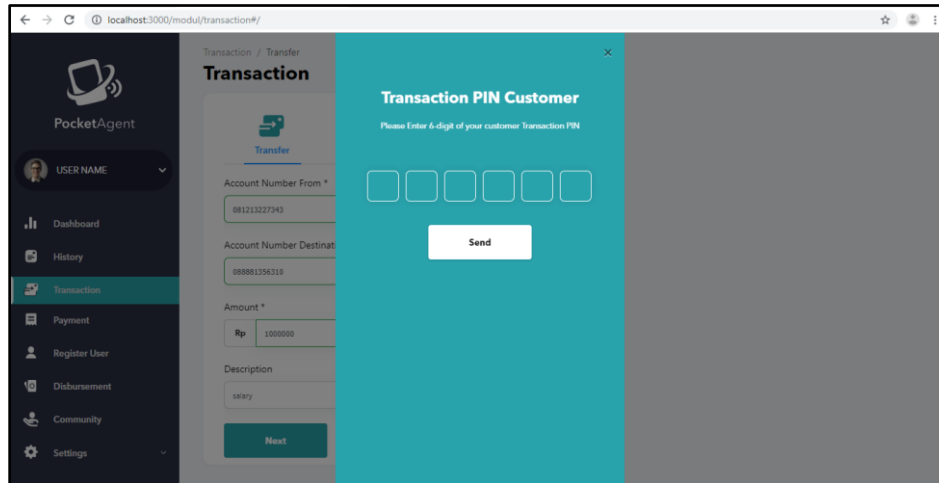
Setelah *user* mengisi *form* pada *transfer* dengan benar, maka fitur akan mengeluarkan *modal confirmation* yang berisi seluruh data yang diinput oleh *user* beserta data keterangan jenis transaksi dan total yang diinput oleh *user* sebelumnya. Pada *modal* ini diharapkan *user* melakukan pemeriksaan ulang terhadap data yang ditampilkan pada *modal Confirmation* untuk memastikan bahwa data yang diinput benar sebelum melanjutkan transaksi. Ketika *user* sudah selesai memeriksa, maka *user* dapat menekan *button* Correct untuk melanjutkan proses *transfer*. *User* juga dapat membatalkan proses transfer dengan menekan *button* X pada ujung kanan atas

modal confirmation. Gambar 3.139 menunjukkan tampilan *modal confirmation* pada transfer.



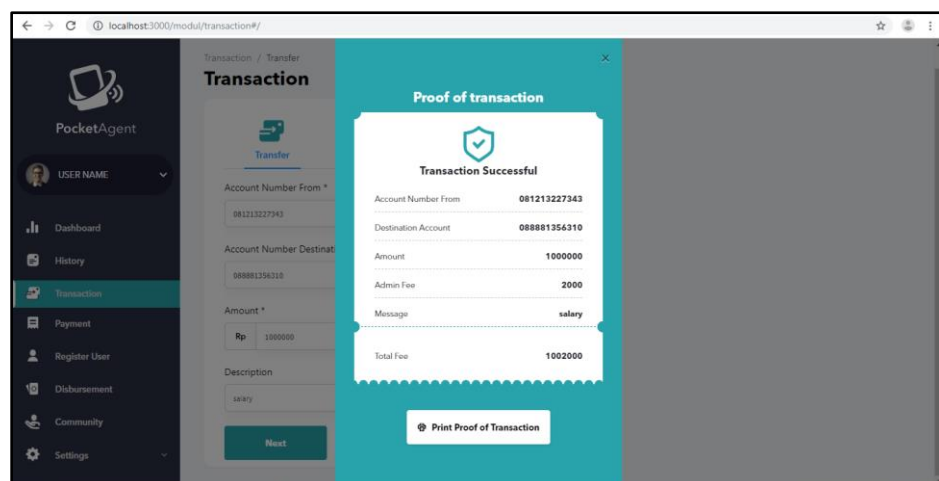
Gambar 3.139 Tampilan Modal Confirmation pada Transfer

Setelah menekan *button correct*, maka *user* akan diminta untuk memasukkan PIN transaksi dengan benar. Secara otomatis aplikasi akan melakukan validasi terhadap PIN transaksi yang diinput oleh *user*. Apabila PIN yang dimasukkan salah, maka fitur akan memberikan *alert notification* merah yang menyatakan PIN Transaksi yang diinput oleh *user* salah dan *user* harus memasukkan PIN Transaksi yang benar untuk melanjutkan proses transfer. Gambar 3.140 menunjukkan tampilan *modal transaction* PIN.



Gambar 3.140 Tampilan Modal Transaction PIN

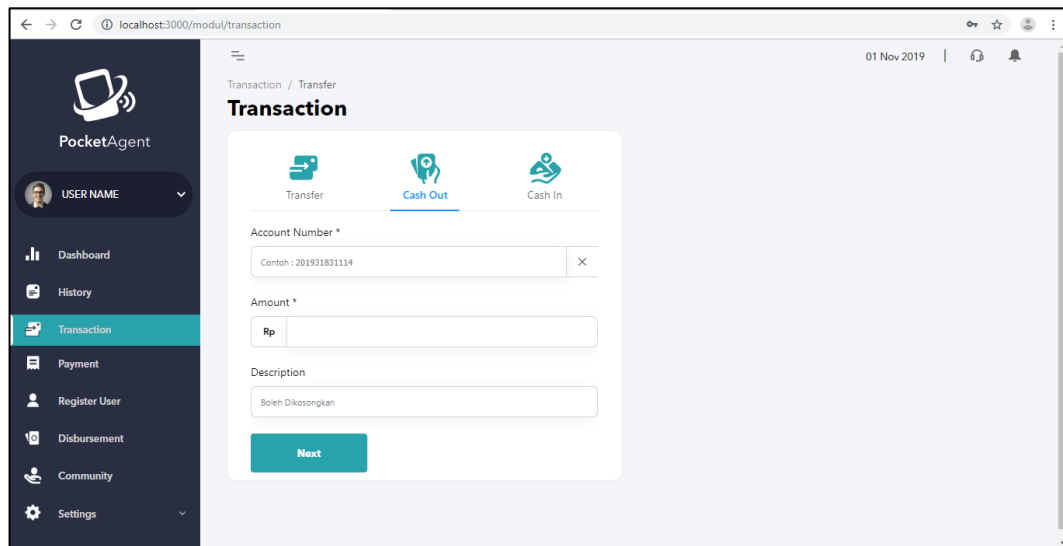
Apabila PIN yang dimasukkan benar, maka fitur akan memberikan *modal* bukti melakukan *transfer* yang menyatakan bahwa proses transfer telah berhasil dilakukan beserta dengan data transaksi yang dimasukkan sebelumnya. *User* diperkenankan untuk mencetak bukti *transfer* dengan menekan *button* Print Proof of Transaction dalam bentuk PDF. Gambar 3.141 menunjukkan tampilan bukti transaksi pada transfer.



Gambar 3.141 Tampilan *Modal* Bukti Transaksi

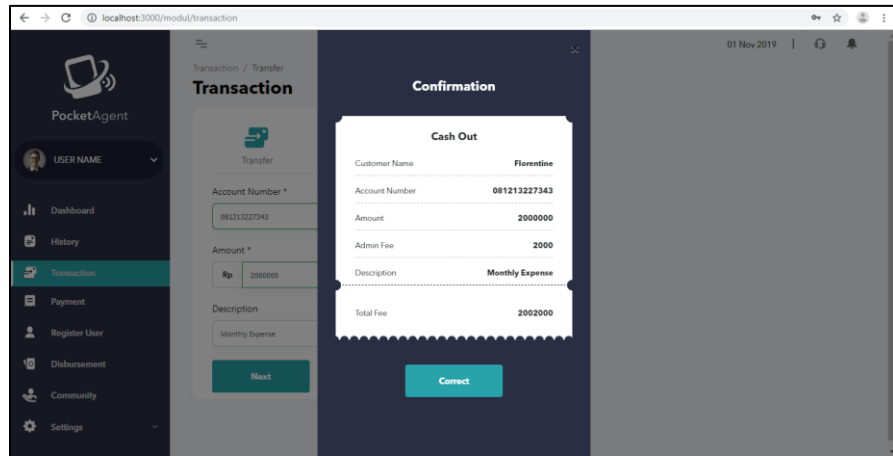
Dalam Menu *Transaction*, *user* juga dapat melakukan melakukan tarik tunai. Gambar 3.142 menunjukkan Tampilan *Cash Out* yang dapat digunakan *user*

untuk tarik tunai. Sama seperti pada *transfer* Fitur *Cash Out* juga terdapat validasi seperti *field required* dan *invalid format*. Dimana *user* harus memasukkan *account number*, jumlah uang yang akan ditarik tunai sesuai dengan *format number*, dan *user* juga dapat menambahkan deskripsi bila diperlukan.

The screenshot shows a web browser at localhost:3000/modul/transaction. The application has a dark sidebar with the 'PocketAgent' logo and a menu including Dashboard, History, Transaction (highlighted), Payment, Register User, Disbursement, Community, and Settings. The main content area is titled 'Transaction / Transfer' and 'Transaction'. It features three tabs: 'Transfer', 'Cash Out' (selected), and 'Cash In'. The 'Cash Out' form includes fields for 'Account Number *' (with a placeholder 'Contoh : 201931831114'), 'Amount *' (with a 'Rp' prefix), and 'Description' (with a placeholder 'Boleh Dikosongkan'). A green 'Next' button is at the bottom of the form.

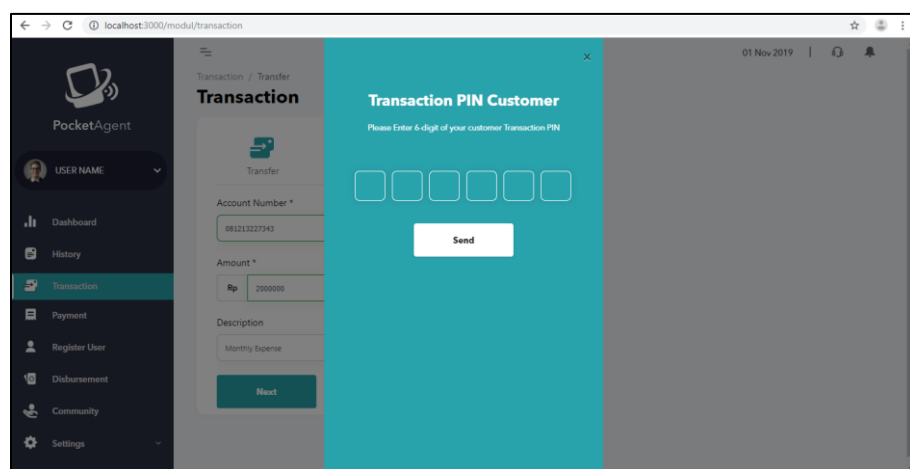
Gambar 3.142 Tampilan Cash Out

Gambar 3.143 menunjukkan *modal* konfirmasi setelah *user* menekan *button next* pada halaman *Cash Out*. *Modal Confirmation* menampilkan semua data yang diinput oleh *user* sebelumnya beserta data yang terdaftar pada aplikasi seperti nama *customer* yang tercantum sebagai pemilik nomor rekening yang diinput oleh *user*. Pada halaman ini, *user* diminta untuk memeriksa kembali akan transaksi yang dilakukan untuk memastikan bahwa data yang diinput oleh *user* dengan benar. Setelah *user* telah merasa data yang dimasukkan dengan benar, maka *user* dapat menekan *button* Correct untuk melanjutkan proses tarik tunai.

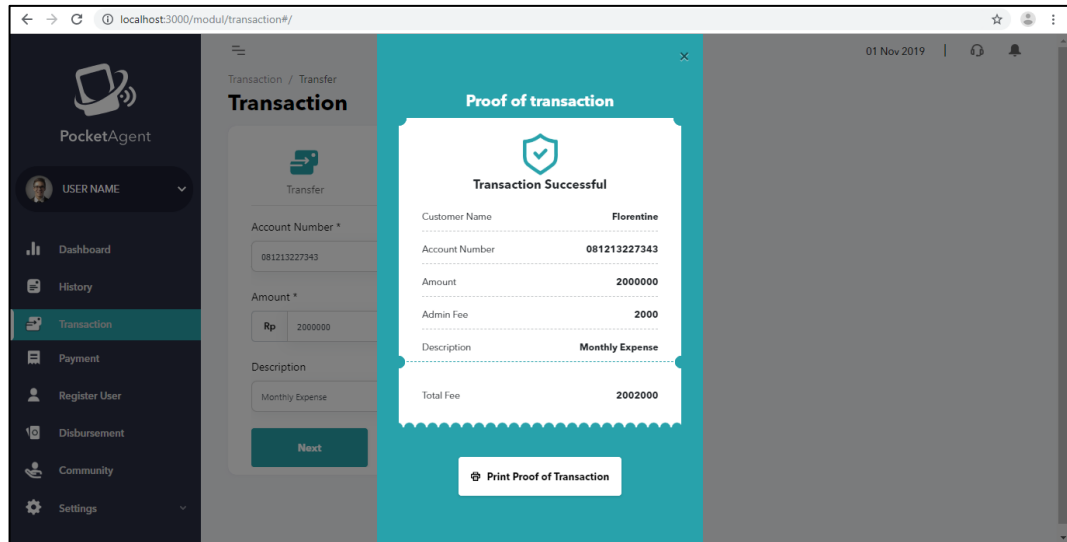


Gambar 3.143 Tampilan Confirmation Modal

Gambar 3.144 menunjukkan tampilan *modal* PIN Transaksi *Customer*, dimana setelah memeriksa data pada *modal confirmation*, maka *user* diminta untuk memasukkan PIN transaksi untuk melanjutkan proses tarik tunai. Secara otomatis aplikasi akan mendeteksi PIN transaksi yang dimasukkan oleh *user*. Bila PIN Transaksi yang dimasukkan oleh *user* salah, maka aplikasi akan mengeluarkan *notification alert* yang menyatakan bahwa PIN transaksi salah dan *user* diminta untuk memasukkan PIN transaksi hingga benar untuk melanjutkan proses tarik tunai. *User* juga dapat membatalkan proses tarik tunai dengan menekan *button X* pada sudut kanan atas pada setiap *modal trasaction*.

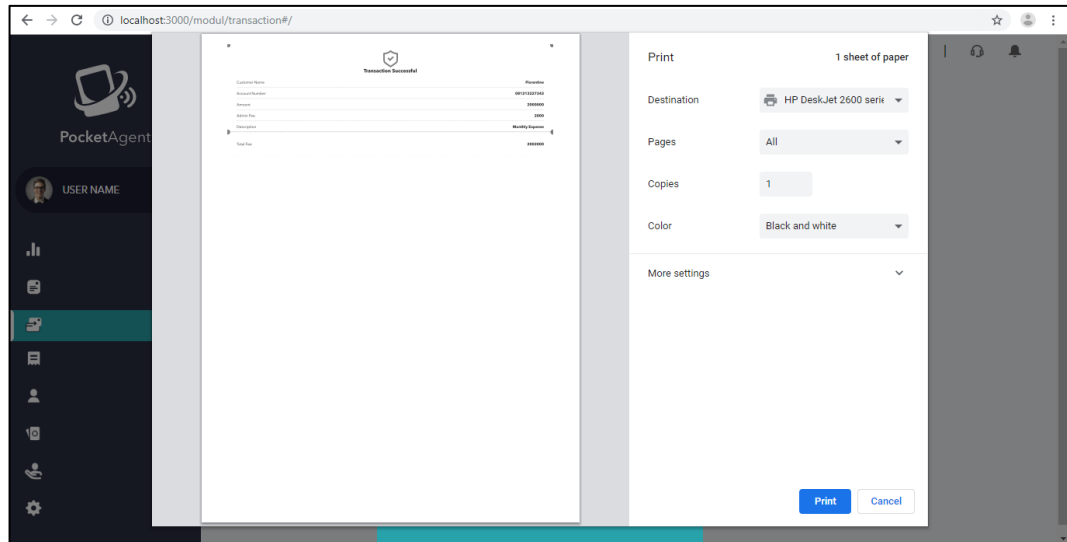


Gambar 3.144 Tampilan *Transaction PIN Customer Modal*



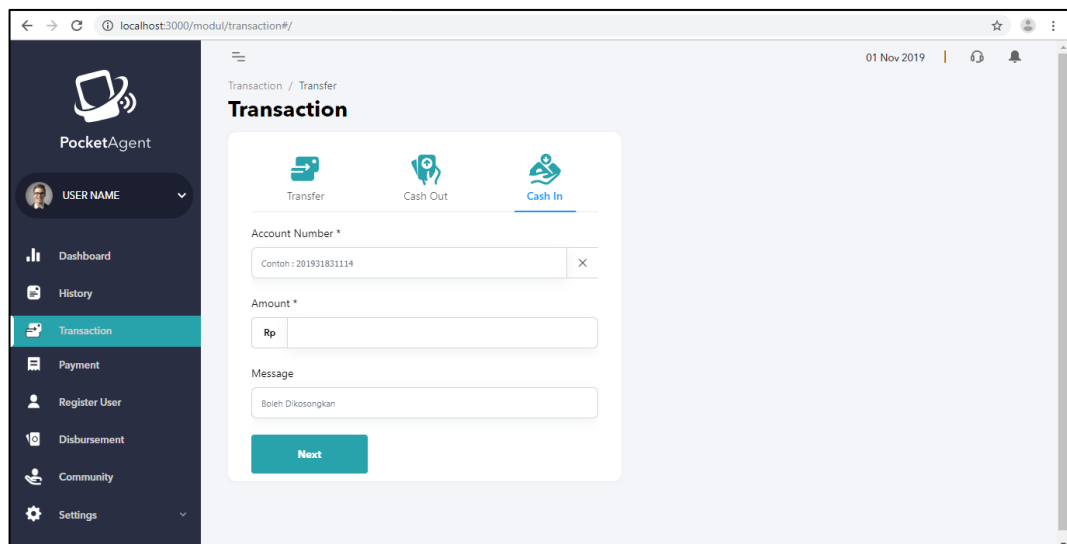
Gambar 3.145 Tampilan *Transaction* Cash Out Invoice

Setelah *user* telah berhasil menginput PIN transaksi dengan benar, maka aplikasi akan mengeluarkan *modal* bukti transaksi sukses dimana *modal* tersebut berisikan data transaksi yang dilakukan oleh *user* dan menyatakan bahwa transaksi yang dilakukan oleh *user* telah berhasil. Gambar 3.145 menunjukkan *modal* transaksi sukses. Pada *modal* bukti transaksi, *user* dapat mencetak transaksi dalam dengan menekan *button* Print Proof of Transaction maka *user* akan mendapatkan sebuah dokumen yang berisikan data transaksi yang dapat langsung dicetak. Gambar 3.146 menunjukkan fungsi print *file* PDF bukti transaksi.



Gambar 3.146 Tampilan Print Function Cash Out

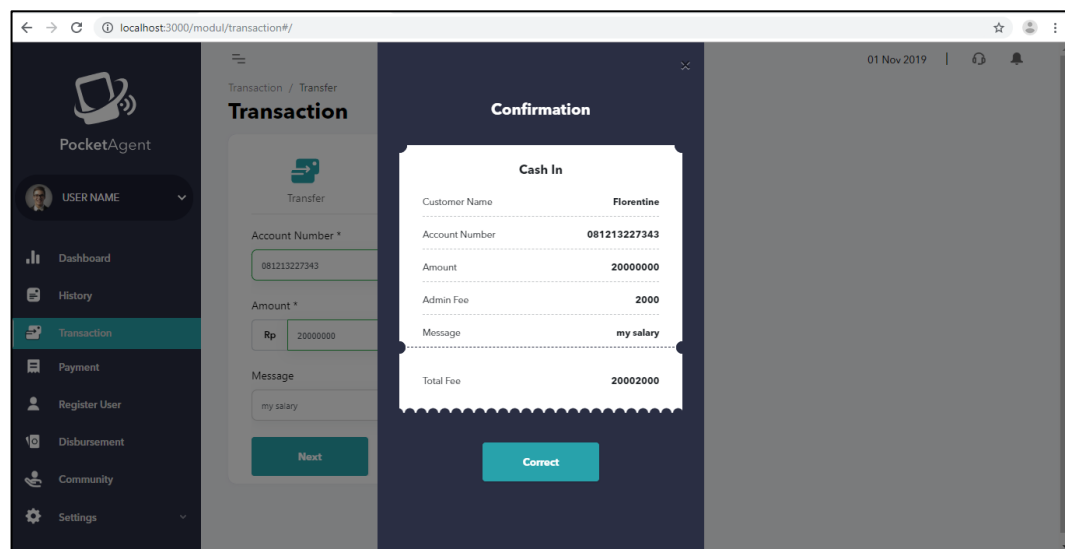
Selain Fitur Tarik Tunai dan Transfer, pada Menu *Transaction* juga terdapat Fitur Setor Tunai atau *Cash In*. Fitur Cash In dapat digunakan *user* untuk melakukan setor tunai dengan nominal tertentu. Gambar 3.147 menunjukkan tampilan Fitur *Cash In*.



Gambar 3.147 Tampilan *Cash In*

Dalam Fitur *Cash In*, *user* diminta untuk memasukkan *account number*, jumlah uang yang akan disetor, dan *user* dapat menambahkan deskripsi jika diperlukan lalu menekan *Button Next*. Sama seperti fitur *transaction* yang lain, Fitur

Cash In juga sudah dilengkapi dengan validasi seperti *field require* dan *invalid format* dimana *user* wajib mengisi semua *form input* pada fitur *cash in* seperti nomor akun dan jumlah nominal yang ingin disetor dalam bentuk *format number* kecuali *form input* pada deskripsi.

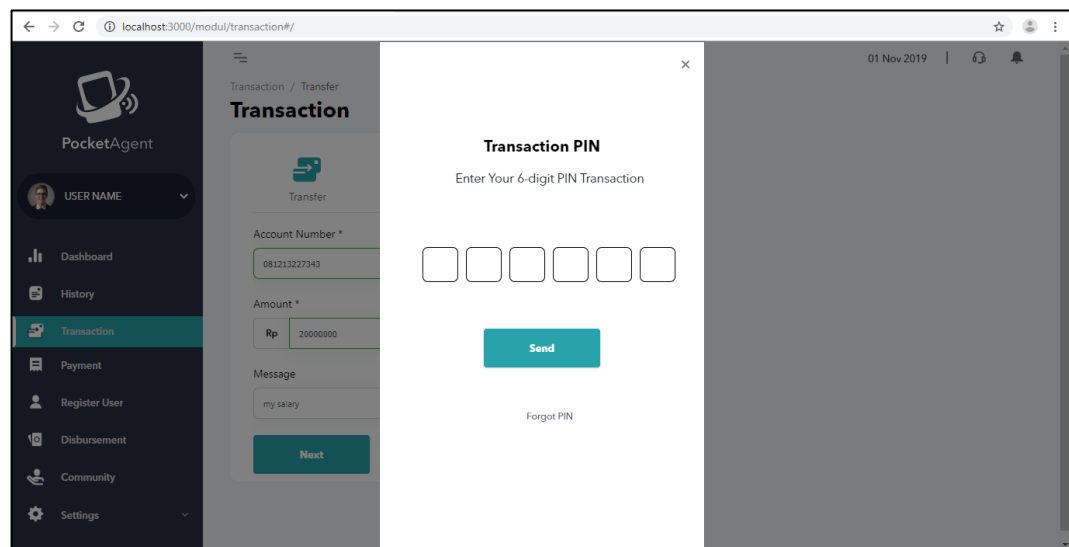


Gambar 3.148 Tampilan *Modal Confirmation Cash In*

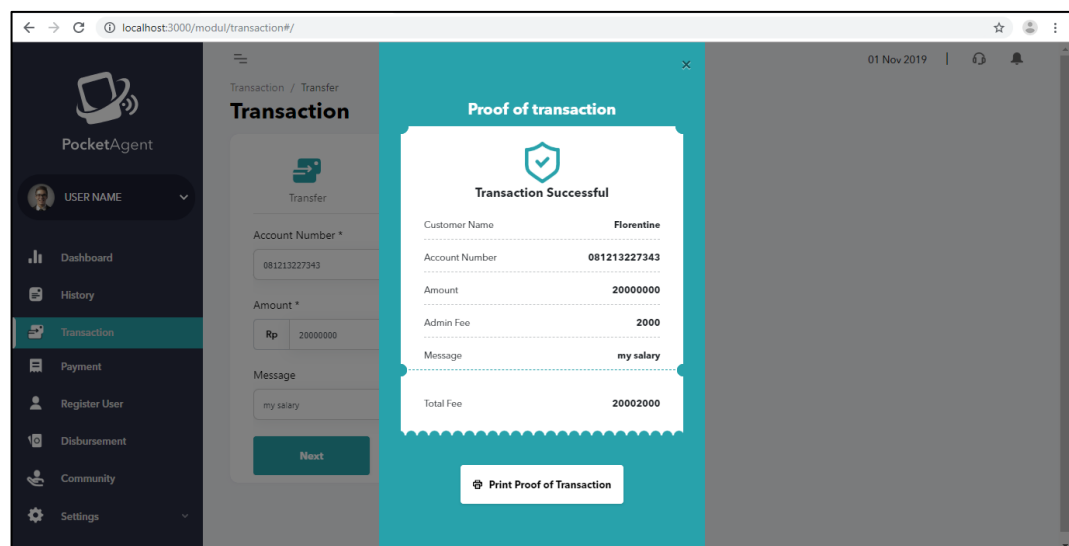
Setelah menekan *Button Next*, *user* diminta untuk memeriksa kembali transaksi pada *Confirmation Modal*. Dalam *Modal Confirmation* sudah terdapat data berdasarkan *input user* sebelumnya yang menjelaskan nama pemilik akun berdasarkan nomor akun beserta dengan nominal dan biaya admin serta deskripsi yang diinput oleh *user* sebelumnya. Gambar 3.148 menunjukkan tampilan *modal confirmation* pada *Cash In*. Setelah memastikan semua data terinput dengan benar maka *user* dapat menekan *button Correct* untuk melanjutkan transaksi setor tunai.

Setelah selesai memeriksa transaksi dengan benar, *user* dapat menekan *Button Next*, maka *user* diminta untuk memasukkan PIN Transaksi. Bila *user* lupa PIN Transaksi, *user* dapat menekan *Button Forgot PIN* dan *user* akan diarahkan ke Fitur *Forgot PIN*. Gambar 3.149 menunjukkan *modal PIN Transaction Customer*

pada *Cash In*

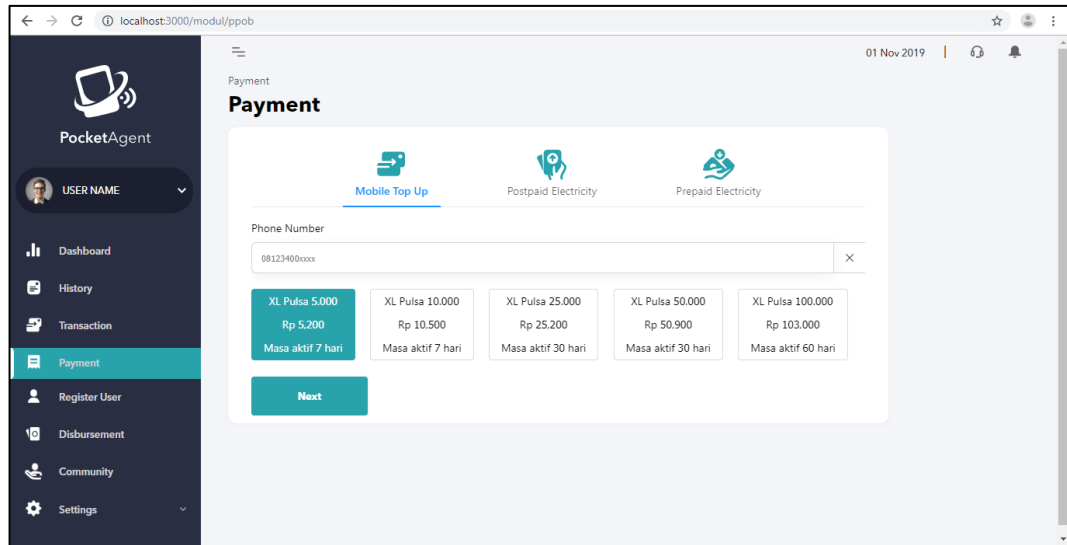


Gambar 3.149 Tampilan *Transaction* PIN Modal



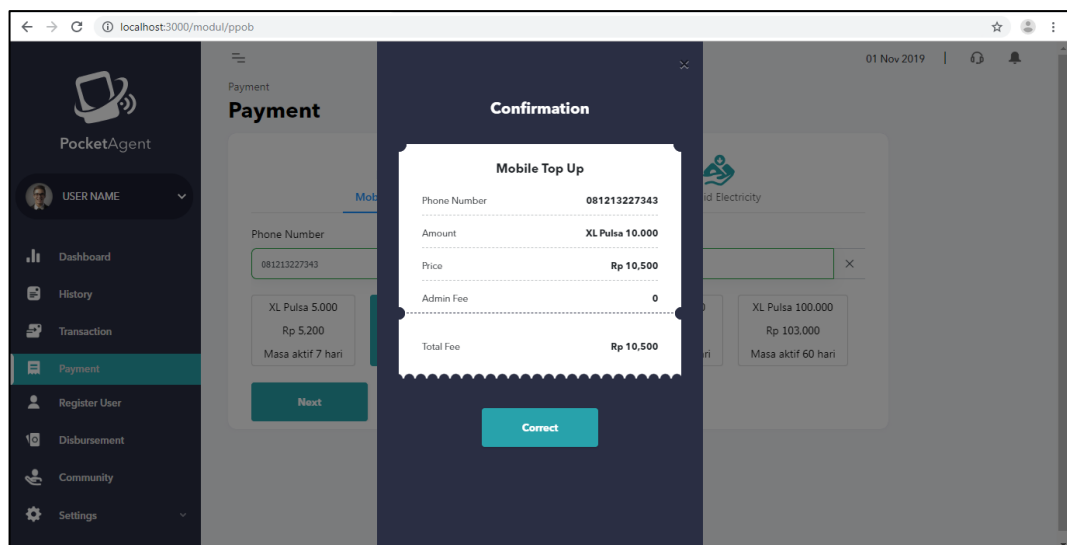
Gambar 3.150 Tampilan Bukti Transaksi *Cash In*

Setelah memasukkan PIN Transaksi, maka *user* diberikan bukti transaksi berdasarkan data input *user* sebelumnya dan *user* diperkenankan untuk mencetak bukti transaksi setor tunai sebagai tanda *user* sudah melakukan setor tunai dengan menekan *button* Proof of Transaction. Gambar 3.150 menunjukkan bukti transaksi pada Fitur *Cash In*.



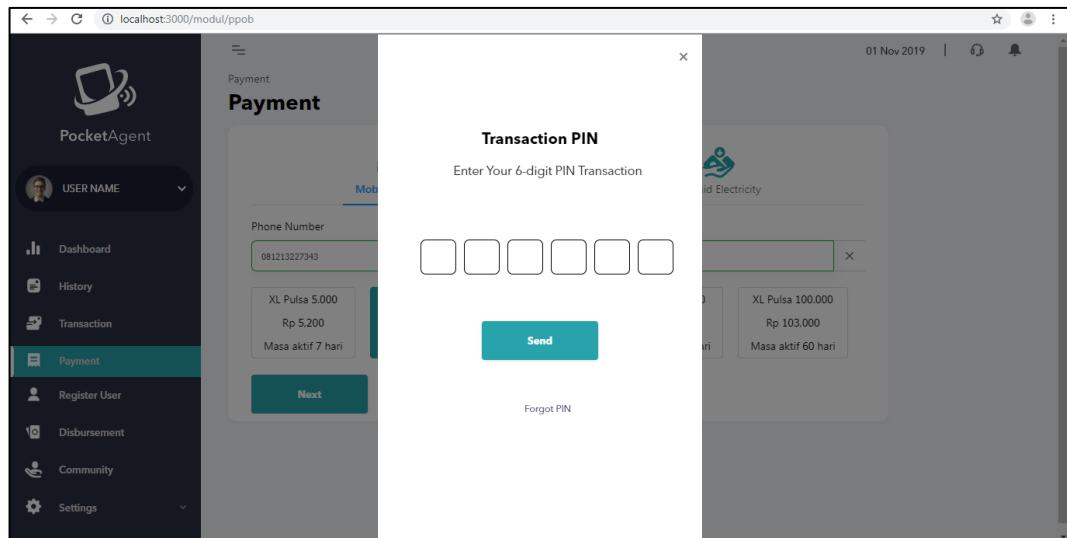
Gambar 3.151 Tampilan *Mobile Top Up*

Gambar 3.151 menunjukkan Fitur *Mobile Top Up* pada Menu *Payment*. Dimana *user* dapat melakukan *top up* pulsa dengan memasukkan nomor telepon dan memilih paket pulsa yang tersedia. Secara otomatis aplikasi akan mengetahui tipe *provider* yang hendak di *top up* oleh *user* beserta dengan paket pulsa yang tersedia beserta harganya. Pada halaman ini *user* diminta untuk memasukkan nomor telepon dan memilih salah satu paket pulsa yang tersedia.



Gambar 3.152 Tampilan *Confirmation Modal Mobile Top Up*

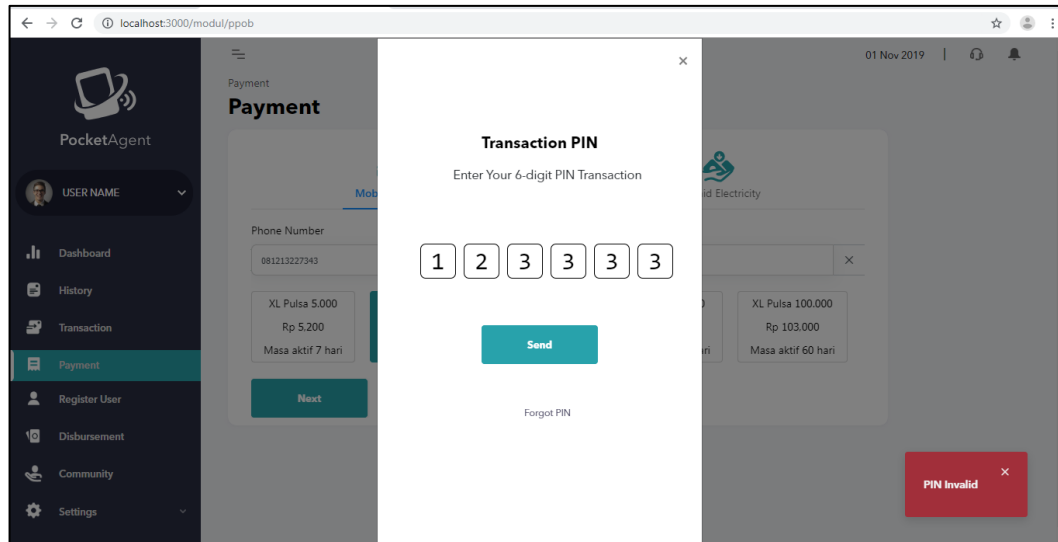
Setelah memilih paket pulsa yang sesuai dengan keinginan, maka *user* diminta untuk memeriksa kembali pembelian *top up* pulsa yang dilakukan berdasarkan data yang dipilih sebelumnya. Gambar 3.152 menunjukkan *modal confirmation* pada *Mobile Top Up*. Setelah memeriksa *user* dapat menekan *Button Correct* untuk melanjutkan transaksi.



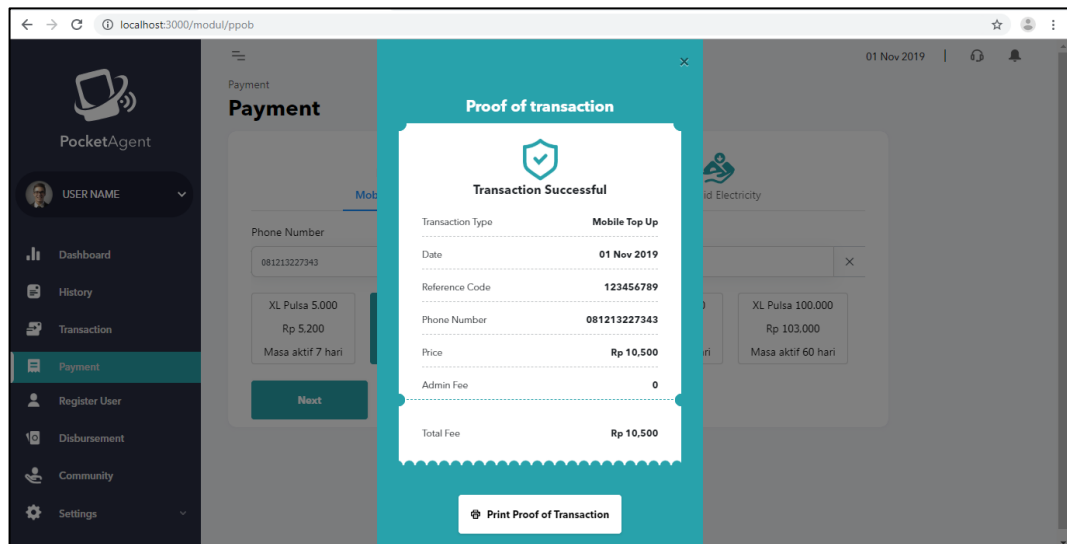
Gambar 3.153 Tampilan *Transaction PIN Modal*

Setelah menekan *Button Correct*, maka *user* akan diminta untuk memasukkan PIN Transaksi. Bila *user* lupa PIN Transaksi, *user* dapat menekan *Button Forgot PIN* dan *user* akan diarahkan ke Fitur *Forgot PIN*. Gambar 3.153 menunjukkan tampilan *modal Transaction PIN*.

Aplikasi secara otomatis melakukan validasi terhadap PIN yang dimasukkan oleh *user*. Apabila PIN yang dimasukkan oleh *user* salah maka aplikasi akan mengeluarkan notifikasi yang menyatakan bahwa PIN yang dimasukkan salah. Gambar 3.154 menunjukkan validasi pada PIN Transaksi, jika PIN Transaksi yang dimasukkan salah.

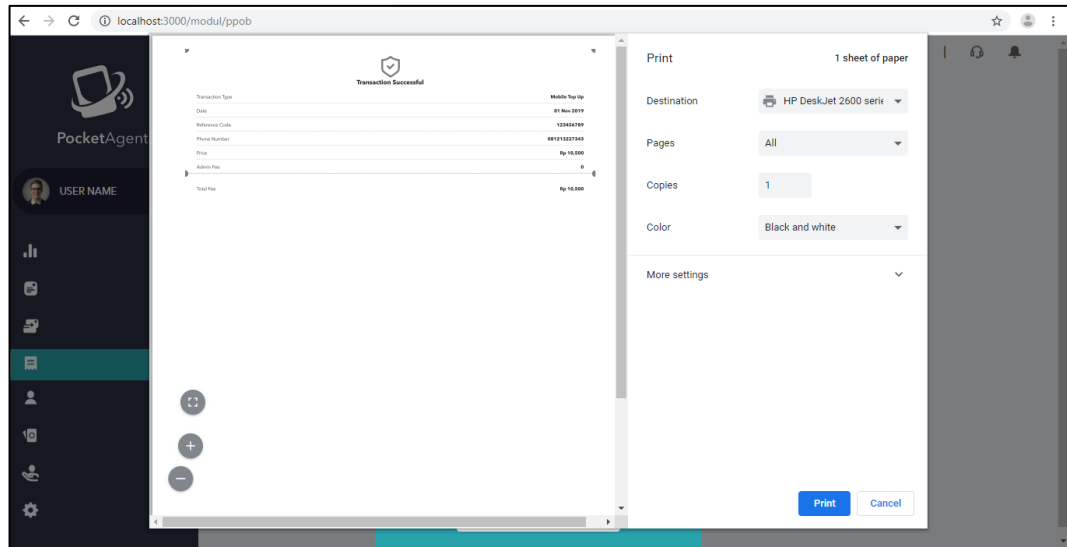


Gambar 3.154 Tampilan Validasi pada PIN Transaksi

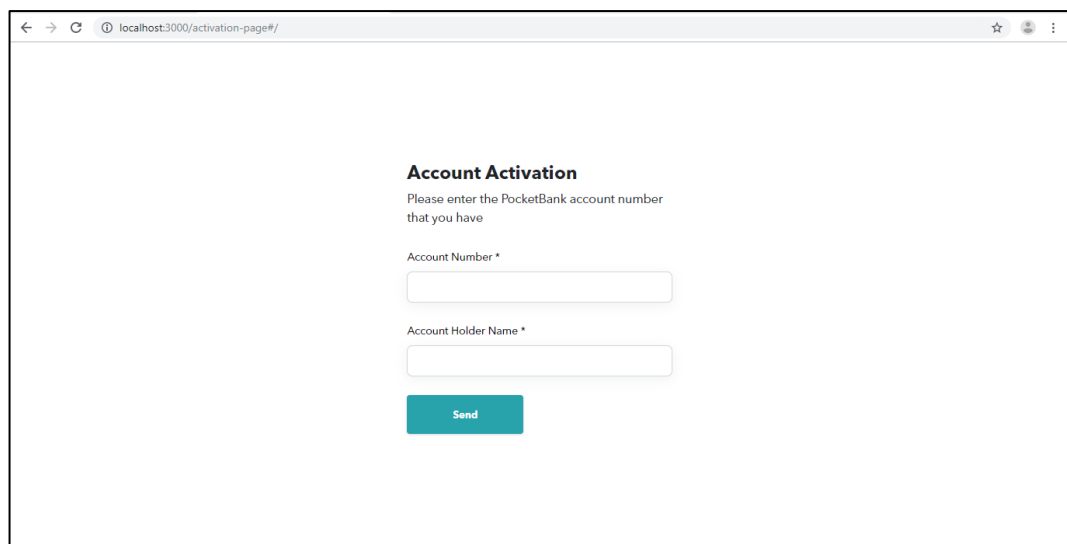


Gambar 3.155 Tampilan Bukti Transaksi *Mobile Top Up*

Setelah selesai memasukkan PIN Transaksi, maka *user* akan diberikan bukti transaksi sebagai tanda transaksi pembelian pulsa yang dilakukan *user* berhasil. *User* juga dapat mencetak bukti transaksi tersebut dengan menekan *Button* Print Proof of Transaction. Gambar 3.156 menunjukkan tampilan *print* pada *Mobile Top Up* dan Gambar 3.156 menggambarkan ketika *user* menekan *buuton* Print Proof Transaction.

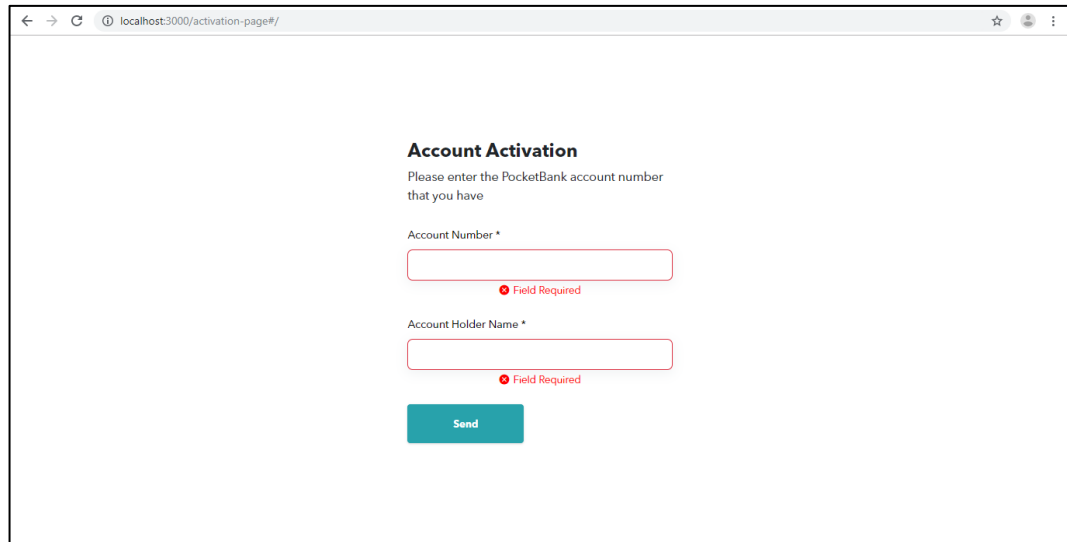


Gambar 3.156 Tampilan Fungsi *Print* pada *Mobile Top Up*



Gambar 3.157 Tampilan Halaman *Activation Account*

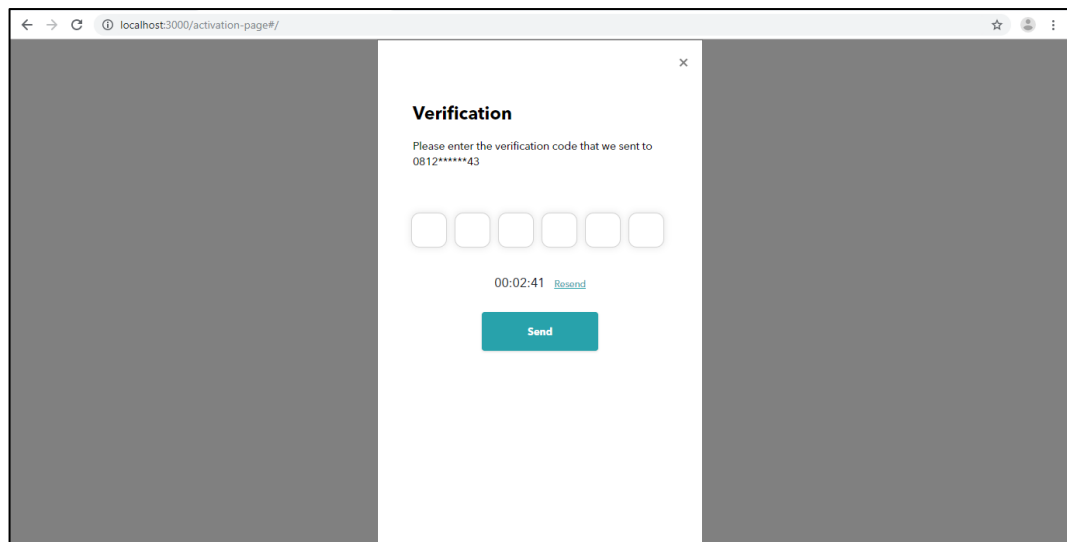
Pada Aplikasi Pocketbank Agent terdapat Fitur halaman Aktivasi Akun yang diletakkan terpisah dari aplikasi. Pada halaman *Activation Account*, *user* dapat melakukan aktivasi akun dengan memasukkan nomor telepon dan nama lengkap. Gambar 3.157 menunjukkan tampilan halaman *Activation Account*.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/activation-page#/'. The page title is 'Account Activation'. Below the title, there is a message: 'Please enter the PocketBank account number that you have'. There are two input fields: 'Account Number *' and 'Account Holder Name *'. Both fields are empty and have a red border with a red dot and the text 'Field Required' below them. At the bottom, there is a teal 'Send' button.

Gambar 3.158 Tampilan Validasi pada *Activation Account*

Pada halaman *activation account* sudah terdapat validasi *field required*. Gambar 3.158 menunjukkan validasi pada halaman *activation account*, dimana *user* tidak dapat melakukan aktivasi jika *user* belum mengisi nomor telepon dan nama lengkap.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/activation-page#/'. A modal titled 'Verification' is open. The modal contains the text: 'Please enter the verification code that we sent to 0812*****43'. Below this text is a row of six empty input boxes. Under the boxes, there is a timer '00:02:41' and a 'Resend' link. At the bottom of the modal is a teal 'Send' button.

Gambar 3.159 Tampilan *Verification Modal*

Setelah *user* memasukkan nomor telepon dan nama lengkap, maka *user* akan diberikan kode OTP ke nomor telepon yang terdaftar sebelumnya. Lalu *user* diminta untuk memasukkan Kode OTP tersebut dengan benar. Gambar 3.159

menunjukkan tampilan modal verifikasi kode OTP pada *Activation Account*.

3.4. Kendala yang Dihadapi

Kendala yang ditemukan selama proses pelaksanaan kerja magang adalah sebagai berikut :

1. Mengalami kesulitan dalam memahami *framework* React Js secara keseluruhan karena penggunaan *framework* yang minim.
2. *Error* dan *bug* yang beruntun pada saat *testing* karena penggunaan *software developement life cycle scrum* yang mengutamakan *bug* dan *error* yang harus dikerjakan lebih dahulu dan pihak perusahaan kurang jelas dalam memberikan validasi dan bahasa yang sesuai dengan standar perusahaan.
3. Terkadang terdapat tidak *support component* React Js yang telah disediakan oleh perusahaan, karena *documentation component* dibuat oleh pihak perusahaan.
4. API yang belum tersedia oleh Tim *Back-End*

3.5. Solusi atas Kendala yang Ditemukan

Berikut ini merupakan solusi atas kendala yang dihadapi selama proses pelaksanaan kerja magang:

1. Mencari semua hal yang kurang dipahami melalui internet dan bertanya pada pembimbing lapangan dan peserta magang lain.
2. Berdiskusi serta meminta pengarahan dari *tester* , pembimbing lapangan, dan peserta magang lain atas *error* dan *bug* yang dihadapi.
3. Melakukan *research* bersama pembimbing lapangan dan *project manager* serta *product owner* terhadap *component* tersebut pada saat *scrum meeting*.
4. Berusaha meminta API kepada Tim *Back-End*.